Maple T.A. User Guide

Copyright © Maplesoft, a division of Waterloo Maple Inc. 2011

Maple T.A. User Guide

Copyright

Copyright © Maplesoft, a division of Waterloo Maple Inc. 2011. All rights reserved.

No part of this book may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means—electronic, mechanical, photocopying, recording, or otherwise. Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this document is furnished under a license agreement and may be used or copied only in accordance with the agreement. It is against the law to copy the software on any medium except as specifically allowed in the agreement.

Trademarks

Maple, Maple T.A., Maplesoft, Waterloo Maple, and Maple T.A. Content Center are all trademarks of Waterloo Maple Inc.

Adobe and Acrobat are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft and Windows are registered trademarks of Microsoft Corporation.

Java and all Java based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maplesoft is independent of Sun Microsystems, Inc.

All other trademarks are the property of their respective owners.

This document was produced using a special version of Maple and DocBook.

Printed in Canada

ISBN 978-1-926902-05-0

Contents

Preface xvii
1 Getting Started 1
1.1 How the System Works 1
User Roles 1
Single Login 1
Class Homepage 1
1.2 Overview: Question Repository, Assignments, and Reports 1
Question Repository 1
Assignments 1
Gradebook 2
1.3 Getting Started 2
Creating a New Class in Maple T.A. 2
Working with Questions and Assignments in Maple T.A. 2
1.4 System Overview 3
The System Login 3
System Homepage 4
The Class Homepage 4
Using the System in the Classroom 5
Getting Help 5
2 Creating and Managing Classes 7
2.1 Class Manager 7
Creating a New Class 7
Registering Users in a Class 10
Shared Classes 13
2.2 Manage Proctors 14
Adding a Proctor 16
Deleting a Proctor 17
2.3 The Class Web Site Editor 17
Uploading Single Files 18
Uploading Multiple Files 18
Viewing or Deleting Files 18
Working with Folders in the Class Web Site 19
3 Creating and Managing Assignments 21
3.1 The Assignment Editor 21
Overview: The Assignment Editor Screen 21
Options in the Assignment Editor 23
Adding New Assignments 23
Copying Assignments 23
Deleting Assignments 24
Editing Assignments 25
Previewing Assignments 26
Printing Assignments 26
Controlling the Sequence of Listed Assignments 26
Overview: Using the Assignment Editor 26
Assigning Partial Credit 27
3.2 Tutorial: Using the Assignment Editor 28
Assignment Editor Step 1: Choose a Name for the Assignment 28
Assignment Editor Step 2: Select Questions 28
Assignment Editor Step 3: Setting Policies 30
Assignment Editor Step 4: Review & Finish 32
3.3 Assignment Types 33

Assignment Types Overview 33
Homework or Quiz Assignments 34
Proctored Exams 35
Anonymous Practice Assignments 35
Study Session and Mastery Assignments 36
Mastery Assignments 36
Study Session Assignments 37
3.4 Details on Using the Assignment Editor 37
Choose Name Tab 37
Select Questions Tab 38
Set Policies Tab 42
Review & Finish Tab 47
4 Working with the Gradebook 49
4.1 Overview of the Gradebook 49
Viewing Scores and Performing Gradebook Searches 49
Generating Grade Reports 49
4.2 Performing Gradebook Searches 50
Setting Search Criteria 50
Setting View Criteria 51
4.3 Searching for Class Grades 51
Viewing Assignment Grades 51
Changing Grades and Adding Comments 53
Changing the Total Points for an Assignment 55
4.4 Searching for Item Statistics 55
4.5 Working with Grade Reports 56
Creating a New Grade Report 57
Viewing a Grade Report 57
Publishing a Grade Report for Students 58
4.6 Exporting Reports and Performance Data 58
4.7 Adding an External Assignment 58
Adding External Grades using a Roster 58
Adding External Grades Individually 59
Modifying External Assignment Grades 59
4.8 Forced Grading 59
5 Course Modules 61
5.1 Course Modules Overview 61
Technical Information: Expected Behaviors of Course Modules 62
5.2 Creating a Course Module 62
Course Module Creation and Shared Class Inheritance of Questions and Assignments 63
5.3 Exporting a Course Module 64
5.4 Importing and Installing a Course Module 64
Requirements 65
Additional Notes 65
Course Module Installation and Shared Class Inheritance of Questions and Assignments 65
5.5 Viewing Details of a Course Module 66
5.6 Adding and Deleting Elements within a Course Module 66
Adding Course Module Contents 66
Deleting Course Module Contents 66
Why Delete Only Part of a Course Module 66
5.7 Deleting a Course Module 67
5.8 Relative Web References in Course Modules 67
6 Administering Maple T.A. 69
6.1 Changing Your Password 69
6.2 Editing Your Profile 69

6.3 Deleting Student Records 69
6.4 Security Issues to Consider 69
Instructor Password 69
Student Account Management 70
Proctored Exams 70
7 Integrating Maple T.A. with an LDAP Server 71
7.1 LDAP Overview for Instructors 71
7.2 Login and Forgotten Passwords with LDAP 71
Login 71
Forgotten Passwords 71
7.3 User Manager - Add Users with LDAP 72
7.4 User Manager - Import with LDAP 72
8 Authoring Methods for Questions 73
8.1 Advanced Instructor Help 73
8.2 Navigating the Question Repository 73
8.3 Overview: Authoring Methods for Question Banks 73
Question Banks 74
Uploading files 74
File Formats: Browsers and Platforms 74
Which Method is Best for my Project? 74
8.4 Project Elements 75
Authoring: When to Use the Question Editor 75
Authoring: When to Use LaTeX 75
Authoring: When to Use Maple T.A. Plain Text Script Files 75
Planning Your Question Bank Project 75
8.5 The Question Editor 76
Creating a New Question or Editing an Existing Question 76
Avoiding System Time-outs during Authoring Sessions 77
Adding and Editing Comments 78
Adding and Editing Algorithms 78
The Algorithm Designer 80
Adding and Editing Information Fields 82
Adding and Editing Hints 83
Adding and Editing a Solution 83
Deleting a Comment, Information Field, Hint, or Solution 84
Editing Question Source Files 84
Redefining Question Types 85
Testing the Grading of a Question 85
Setting Answer Format in Numeric Questions 85
Setting Answer Tolerance in Numeric Questions 86
Images and Reference Files in Questions 87
Flash Movies and Java Applets in Questions 89
Graph Plotting Applet in Questions 90
Using a Maple Repository 90
8.6 Question Authoring in the Question Editor 92
Online Authoring in the Question Editor 92
Clickable Image Questions in the Question Editor 93
Essay Questions in the Question Editor 94
Free Response (Fill-in-the-Blank) Questions in the Question Editor 95
Maple-graded Questions in the Question Editor 96
Matching Questions in the Question Editor 98
Mathematical Formula Questions in the Question Editor 98
Multipart Questions in the Question Editor 100
Multiple Choice Questions in the Question Editor 101

Multiple Selection Questions in the Question Editor 101
Numeric Questions in the Question Editor 102
Palette-based Symbolic Editor Questions in the Question Editor 103
Question Designer Questions in the Question Editor 104
True-or-False Questions in the Question Editor 105
Creating Pop-Up References in Questions 105
Questions with Labeled Images 106
Text Formatting in Questions 107
Tutorial: Maple Syntax Questions with Matrices 107
Tutorial: Working with Multipart Questions 112
8.7 Question Repository 114
Searching for Questions in the Question Repository 114
Organizing Questions in the Question Repository 115
Deleting Questions from the Question Repository 115
Previewing a Question 116
Exporting Work to Your Hard Drive 116
8.8 Sharing Questions 117
8.9 Question Banks 118
What is a Question Bank? 118
Topic Structure within Question Banks 118
Opening a Saved Question Bank File 120
Saving a Question Bank 120
Saving to a File versus Reinstalling a Question Bank 121
Creating and Opening Modules 121
Question Fields 122
Question Modes (Question Types) 124
Required Question Elements 125
The name Field: Using Question Descriptions 126
The answer Field 127
The info Field 128
Hints, Coaching, and Performance Feedback 128
Comments and Performance Feedback 129
Using Complete Solutions (Study Session Assignments) 129
Special Characters in Questions 130
9 Question Types 131
9.1 Question Types Available in the System 131
9.2 Math Question Types 132
Math Question Types - Overview 132
Math Question Types Comparison Table 133
Blanks [formula] 134
Chemistry 135
Constants of Integration Questions 136
Equation 137
Formula 137
Formula - Form Variant 138
Formula List 139
Formula with Units 139
Maple-graded Question Type - Overview 140
Maple-graded Formula 142
Maple Syntax 143
Matrix 144
Multi Formula 145
Multipart Formula 146
Numeric 146

Questions with Plots 148
Restricted Formula 150
Vectors of Formulas 150
9.3 Applet Questions 151
Question Mode: Applet 151
Applet Question Authoring 152
9.4 Blanks Questions 153
9.5 Clickable Image Questions 154
Question Mode: clickable image 154
9.6 Ungraded Essay Questions 155
Question Mode: essay 155
9.7 Flash Questions 155
Question Mode: Flash 155
Flash Question Authoring 155
9.8 Free Response and Fill-in-the-Blank Questions 156
List Questions and Blanks Questions 156
Rules for Grading Free Response Questions 158
9.9 Key Word or Phrase Questions 160
Question Mode: key words 160
9.10 List Questions 160
Summary of Question Behaviors 160
9.11 Matching Questions 161
Question Mode: Matching 161
9.12 Multipart Questions 162
Multipart Questions 162
Multipart Questions - Adding or Editing Parts 163
Multipart Questions: Partial Credit 163
Multipart Questions - Sequencing Parts 163
9.13 Multiple Choice Questions 163
Question Mode: multiple choice; non permuting multiple choice 163
9.14 Multiple Selection Questions 164
Question Mode: multiple selection; non permuting multiple selection 164
9.15 Palette Questions 165
Question Mode: palette 165
9.16 Question Designer Questions 166
Authoring with the Question Designer 167
Formatting Tips 168
Question Types and Properties 169
Edit/Insert Response Area 170
Response Area Properties 171
Insert Algorithm 176
9.17 Short Phrase Questions 177
Question Mode: short phrase 177
9.18 True-or-False Questions 177
Question Mode: True False 177
9.19 Advanced Question Types 178
10 Authoring Mathematical Questions 179
10.1 Authoring Mathematical Questions 179
Fundamental Capabilities 179
Mathematics Question Types 179
10.2 Basic Math Syntax in the System 180
Writing Mathematical Expressions 180
Variable Names 180
Operator and Function Syntax 181

10.3 Understanding the Math Capabilities in the System 181
10.4 Student Responses and Grading 182
Student Responses 182
Controlling Answer Format in Numeric Questions 182
Controlling Answer Tolerance 183
Setting a Margin of Error in Non-numeric Questions 184
Significant Digits in Maple Questions 185
Using Maple Code to Prevent Cheating in Maple Syntax Questions 185
Entering Chemistry Expressions 186
Plotting a Student Response 187
10.5 Using Formula Syntax 188
10.6 Using the Equation Editor to Create Formatted Math Expressions 188
Editing with the Equation Editor 188
Equation Editor Palette 190
Equation Editor Requirements and Syntax 192
10.7 Using MathML to Create Formatted Math Expressions 196
Inline Display of Formatted Math Expressions 196
MathML Authoring with MathType 197
MathML in the Questions 198
Simplified Math Expressions 199
10.8 Plotting Using Maple 201
Displaying a Maple Plot 201
10.9 Mathematical Functions & Operations 204
10.10 Using a Table of Units 205
Default Table of Physical Unit Equivalents 205
Custom Table of Equivalent Units 211
10.11 Tutorial: Displaying Answers vs. Formulas for Answers 212
Displaying the answer Field Formula: 212
Displaying the Calculated Final Answer: 212
Displaying Both the Formula for the Answer and the Final, Calculated Answer: 213
10.12 Tutorial: Formatting Integers and Reals 214
10.13 Tutorial: Setting Margin of Tolerance in Non-numeric Questions 215
Absolute Errors 215
Percentage Errors 216
Algorithmic Variables 216
Using the ? Operator in Complex Expressions 216
10.14 Overview of Algorithmic Question Capabilities 217
10.15 How Does an Algorithmic Question Work in an Assignment? 218
Authoring 218
Assignment Creation 218
Assignment Use 218
10.16 Working with Variable Data 218
Types of Variables 218
Functions within Algorithms 219
Inline Variable Expressions 220
Rules for Naming Variables 221
10.17 Generating Random Numbers in Questions 221
10.18 Functions and Arguments 222
condition:x 222
decimal(n, x), sig(n, x), int(x) 223
eq(a, b), ge(a, b), le(a, b), ne(a, b) 223
fact(n) 224
frac(a, b) 224
gcd(a, b) 225

gt(a, b), lt(a, b) 225
if(a, b, c) 225
indexof(k, a, b, c, d, ...) 225
java(cn, a, b, c, d, ...) 225
lsu(n, x) 226
maple(text) 226
mathml(f) 226
max(a, b, c, d, ...), min(a, b, c, d, ...) 227
not(a) 227
numfmt(fmt, x) 227
plotmaple(text) 227
rand(m, n), rand(m, n, k) 228
Random Integer Generation range(n), range(m, n), range(m, n, k) 229
rank(n, a, b, c, d, ...) 230
Random Integer Generation rint(n), rint(m, n), rint(m, n, k) 230
Statistical Operations 231
strcat(a, b, c, d, ...) 232
sum(varname, start, stop, expr) 232
switch(n, a, b, c, ...) 232
10.19 Mathematical and Logical Operators in Algorithms and Answers 233
Unreserved Operations 233
Reserved Operations 233
10.20 Question Chaining Variables 233
Example 234
10.21 Setting Plot Options 235
2-D Plot Statement Options 235
3-D Plot Statement Options 237
10.22 Details on Using Algorithmic Variables 239
The Answer Field Evaluation and Requirements 239
Controlling Display of Numbers within Questions 239
Variables within Variable Statements 239
Displaying the $ (Dollar Sign) in Questions 240
Displaying the Backslash Symbol in Questions 240
Text Strings in Variable Statements 240
10.23 Tutorial: Algorithmic Question 241
10.24 Tutorial: List-based Variables 242
Example 242
10.25 Tutorial: Maple-based Algorithmic Question Authoring 244
Using a Text or HTML Editor 244
Using the Question Editor 244
Tutorial 1 244
Tutorial 2 246
11 Authoring Questions in LaTeX 247
11.1 LaTeX Authoring Overview 247
Requirements 247
Authoring Process 247
11.2 LaTeX Question Bank Document Structure 247
11.3 Question Structure 248
Available Fields in LaTeX Questions 248
11.4 LaTeX Style Sheets 249
11.5 Basic Question Types in LaTeX 250
Blanks Questions in LaTeX 250
Clickable Image Questions in LaTeX 252
Free Response Text-based Questions in LaTeX 253

Matching Questions in LaTeX 254
Multipart Questions in LaTeX 255
Multiple Choice Questions in LaTeX 256
Multiple Selection Questions in LaTeX 257
True-or-False Questions in LaTeX 258
11.6 Advanced Question Types in LaTeX 259
Graph Plotting in LaTeX Questions 259
Graph Sketching Questions in LaTeX 260
Including Graphic References in LaTeX Questions 262
Labeling Graphics in LaTeX Questions 263
11.7 Tutorial: Using Algorithms in LaTeX Questions 264
Algorithmic Questions 264
11.8 Math Question Authoring using LaTeX 267
Other Math Question Features 267
Mathematical Fill in the Blanks - [formula] option 268
Constants of Integration Questions in LaTeX 269
Equation Questions in LaTeX 270
Formula List Questions in LaTeX 271
Formula Questions in LaTeX 271
Maple-graded Questions in LaTeX 272
Matrix Questions in LaTeX 274
Multi Formula Questions in LaTeX 274
Numeric Questions in LaTeX 276
Restricted Formula Questions in LaTeX 279
Coordinates of Points and Vectors Questions in LaTeX 280
12 Plain Text Script Authoring 283
12.1 Using Script Files to Author Questions 283
12.2 Library of Example Scripts 283
Creating Algorithmic Graphic Labels 284
Algorithmic Mathematical Formula Question Example Script 284
Algorithmic Multiple Selection Question Example Script 284
Answer Tolerance Example Script 285
Applet Questions 288
Clickable Image Question Example Script 288
Essay Question Example Script 289
Fill-in-the-Blank Question Example Script 289
Graph Plotting Example Script 290
Graph Sketching Question Example Script 290
Key Word or Phrase Question Example Script 291
Labeling Graphics in Questions Example Script 291
List Question Example Script 292
Maple-graded Formula Question Example Script 293
Maple Syntax Question Example Script 294
Multipart Question Example Script 295
Multiple Choice Question Example Script 296
Multiple Selection Question Example Script 296
Authoring Questions with Embedded Plots Example Script 297
Question Designer Question Example using Various Embedded Questions 298
Short Phrase Question Example Script 300
True-or-False Question Example Script 300
A Additional Information 303
A.1 Copyrights and Trademarks 303
Copyrights 303
Trademarks 303

A.2 System Requirements 303
Client Recommended Configuration: Instructors, Students, and Proctors 303
Windows - 32-bit 304
Windows - 64-bit 304
Macintosh 304
Linux - 32-bit 304
Linux - 64-bit 304
Internet Access 304
Index 305

List of Figures

Figure 1.1: System Homepage with Classes I Am Instructing 4
Figure 1.2: Class Homepage with Assignment List 4
Figure 2.1: Inherit Content from a Featured Class 8
Figure 2.2: Proctor Tools 15
Figure 2.3: Add a Proctor to Your Class 16
Figure 2.4: Web Site Editor 17
Figure 3.1: Assignment Editor 21
Figure 3.2: Options in the Assignment Editor 23
Figure 3.3: Add Questions to Assignment 29
Figure 3.4: Select Questions: Specify Order and Weighting of Questions 29
Figure 3.5: Setting the Assignment Type 30
Figure 3.6: Feedback Options 31
Figure 3.7: Assignment Properties 31
Figure 3.8: Scheduling 31
Figure 3.9: Visibility 32
Figure 3.10: Set Criteria for Assignments 32
Figure 3.11: Review Assignment Details 33
Figure 3.12: Policies for Mastery Assignments 36
Figure 3.13: Assignment Annotations 41
Figure 5.1: Create, Import, Export, or Delete Course Modules 61
Figure 8.1: Algorithm Editor 79
Figure 8.2: Create a Random Number Variable 80
Figure 8.3: Specify Acceptable Answer Format in a Numeric Question 85
Figure 8.4: Specify Answer Precision and Tolerance in a Numeric Question 86
Figure 8.5: Uploading and Linking a Maple Repository 91
Figure 8.6: Question Designer Question 104
Figure 8.7: Defining a Matrix 108
Figure 8.8: Defining a Simple Matrix Question 109
Figure 8.9: Using the Matrix Palette 110
Figure 8.10: Invertible Matrix Question 112
Figure 8.11: Search Results in the Question Repository 114
Figure 9.1: Constants of Integration Question 136
Figure 9.2: Equation Question 137
Figure 9.3: Input Field for a Formula Question 137
Figure 9.4: Input Field for a Formula List Question 139
Figure 9.5: Input Field for a Multi Formula Question 145
Figure 9.6: Input Field for a Multipart Formula Question 146
Figure 9.7: Input Field for a Vector of Formulas Question 151
Figure 9.8: Three Varieties of Blanks Questions: Drop-Down List, Text Box, and Formula Box 153
Figure 9.9: Clickable Image Question 154
Figure 9.10: Demonstration of Various Free-Response and Fill-in-the-Blanks Questions 157
Figure 9.11: Question Designer Screen 167
Figure 9.12: Tabs for Specifying Feedback, Algorithms, and Additional Question Info 167
Figure 9.13: Access the Equation Editor in the Question Designer 168
Figure 9.14: Equation Editor Palettes 169
Figure 9.15: Add Response Area in the Question Designer 170
Figure 9.16: List Response Area in the Question Designer 172
Figure 9.17: Maple-graded Response Area in the Question Designer 173
Figure 9.18: Multiple Choice Response Area in the Question Designer 174
Figure 9.19: Numeric Response Area in the Question Designer 176
Figure 10.1: Equation Editor 189

Figure 10.2: MathType Translator to MathML 197
Figure 10.3: Plot with Algorithmic Variables 202

List of Tables

Table 1.1: System Modules Overview 3
Table 3.1: Assignment Attributes 33
Table 4.1: Assignment Summary 51
Table 4.2: Item Statistics 55
Table 8.1: Maple-graded Question with Partial Credit 97
Table 8.2: Expression Types Available in Mathematical Formula Questions in the Question Editor 99
Table 8.3: Examples of HTML Tags to Use for Formatting Text 107
Table 8.4: Question Repository: Available Actions 115
Table 8.5: Question Fields 122
Table 8.6: List of Question Types and Corresponding modes 124
Table 8.7: Answer Field Requirements by Question Type 127
Table 9.1: Math Question Types Comparison 133
Table 9.2: Rules and Keys for Entering Chemistry Expressions 135
Table 10.1: Guidelines for Entering Chemistry Expressions 186
Table 10.2: Keyboard Shortcuts for the Equation Editor 189
Table 10.3: Defining Palettes 193
Table 10.4: Options for the mathml Function 200
Table 10.5: Mathematical Functions and Operators 204
Table 10.6: Hyperbolic and Inverse Hyperbolic Functions 204
Table 10.7: Maple Notation 204
Table 10.8: Formatting Numbers with $m 214
Table 10.9: Random Number Generators 221
Table 10.10: Maple Randomization Commands 222
Table 10.11: Options for GIF and JPEG Plots 228
Table 10.12: Mathematical Functions that Students Can Use 233
Table 10.13: Mathematical Functions Reserved for Instructor Use 233
Table 10.14: Options for 2-D Plots 235
Table 10.15: Option for 3-D Plots 237
Table 11.1: Example LaTeX Document Structure 248
Table 11.2: Basic LaTeX Question Structure 248
Table 11.3: Arguments and Option for a Graph Plotting Applet 259
Table 11.4: Arguments and Option for a Multigraph 259
Table 11.5: Arguments and Option for a Sketch 260
Table 11.6: Basic Sketch Criteria 260
Table 11.7: Statements for Additional Sketch Criteria 260

Preface

Overview of the Instructor Help

Release 7

Maple T.A. has three components:

- **Question Repository** - collection of questions
- **Assignments** - tests containing questions from the question repository with policies that control their form and availability
- **Gradebook** - results of student assignments

For more information, see the *Overview: Question Repository, Assignments, and Reports (page 1)*.

The **Class Homepage** is the starting point for instructors, students, and proctors. To learn more about the Class Homepage, see *How the System Works (page 1)*.

The Instructor Help System in Maple T.A. includes information on:

Instructor Tasks: Chapters 1-7	These chapters provide guidelines for creating assignments, querying the Gradebook, and using the system tools.
Advanced Authoring: Chapters 8-12	These chapters offer more advanced authoring help, providing descriptions of supported question types and guidelines for creating questions.

The Maple T.A. **User Guide** is available for instructors from the help menu. Select **Help>User Guide>Download PDF**. You can also visit the Maple T.A. Content Center at http://www.maplesoft.com/tacontent to download additional Maple T.A. resources.

What's New in Maple T.A. 7

Release 7

Maple T.A. now offers new gradebook features, including the ability to create customized grade reports that reflect your grading policy. This release also includes enhanced support for Maple-graded questions.

New Gradebook Features

- From a single screen, instructors can view all responses to any questions in an assignment.
- Instructors can create grade reports by grouping assignments and assigning weights to the groups. Options include designating extra credit and dropping the lowest grades. These reports can be saved and published for students. For more information, see *Working with Grade Reports (page 56)*.
- For any assignment in a parent class, the instructor can access the Gradebook data for all child classes.
- Class grades on external assignments can be batch-imported from a comma delimited file.

Enhancements to the Maple-graded Question Type

- Maple Syntax questions now support symbolic response entry, with an expanded palette that includes integrals, derivatives, and matrices.
- Students can build a matrix response in a natural way and instructors have full control over how the matrix is graded.

For more information, see *Tutorial: Maple Syntax Questions with Matrices (page 107)*.

See Also:

Copyrights and Trademarks (page 303)

1 Getting Started

1.1 How the System Works

User Roles

Each user in the system has a default role as assigned by the administrator: **Administrator**, **Instructor**, **Proctor**, **Student**, or **Guest**. Users can have different roles in different classes. For example, an instructor can be a student in one class, with student privileges in that class, and he or she will have instructor privileges in the class he or she is instructing.

Single Login

All users log in to the system through the same login screen. Once in the system, users have access to different menu items depending on their role. Users can register in classes, or access the class homepage of classes they are instructing, proctoring, or taking as a student.

Class Homepage

To use the system for homework management and online testing in your class, instructors begin by creating a class. Once your class is set up, students access assignments, quizzes, proctored tests, and view their results through your class homepage.

Students can register in your class if you allow for open enrollment; otherwise you can add students by uploading a student roster or through the user manager.

Instructors access the Class Homepage to:

- Use pre-existing question banks
- Create, publish, and manage assignments
- Manage student interactions
- Access a **Gradebook** that records student work and assignment results

1.2 Overview: Question Repository, Assignments, and Reports

The variety of question types (including sophisticated Math and Science questions that use the Maple™ computer algebra system), range of assignment modes, powerful tools, and reports the system can generate make Maple T.A. an ideal productivity enhancement for instructors, while giving students an individualized learning experience.

Question Repository

Using Maple T.A., you create, store, and edit questions in the question repository. Questions are the basis for self-scoring assignments. You can easily select questions from the question repository to create an assignment.

Assignments

Maple T.A. assignments automate common learning activities that involve questions and problems. You can create assignments for a range of purposes, including:

- Self-study practice
- Mastery
- Homework & Quizzes
- Proctored exams requiring proctor validation

When an assignment is created, you specify assignment access, grading parameters, and feedback options. After determining the content (questions), rules, and policies, you *publish* an assignment to the class web site for use by students. Students can review their results of any previously completed assignments.

Gradebook

After students complete an assignment, you can view the results in a **Gradebook** or create downloadable performance reports that can be viewed in third-party gradebook programs or Microsoft® Excel.

With Maple T.A., you can generate comprehensive performance reports for individual students, individual class sections, or multiple class sections. You can also perform statistical item analysis of questions.

1.3 Getting Started

Creating a New Class in Maple T.A.

Your first step is to log in to the Maple T.A. system using the username and password that your system administrator or course coordinator has provided. Once you are logged into the system, you can create a new class (if you have sufficient privileges). You can upload a student roster or set open registration so that system users can enroll themselves in your class. For details on creating a class, see *Creating a New Class (page 7)*.

Working with Questions and Assignments in Maple T.A.

Question Repository

With Maple T.A. you create self-scoring assignments with questions drawn from the question repository. The system's assignments can help automate common learning activities that involve questions and problems in classes such as practice and proctored exams, tutorial sessions, homework, and quizzes. The system can report on individual student performance, performance across a class section or across multiple sections of a class, and statistical item analysis of questions.

The system can be configured to enable easy sharing of class content and assignments across sections of large classes.

For more information, see *Creating and Managing Assignments (page 21)*.

Assignments and Results

When you create an assignment, you specify the rules and policies for assignment access and due dates, grading parameters, feedback to students, and whether student results are recorded in the Maple T.A. **Gradebook.** After students complete an assignment, you can view results in a **Gradebook** or create downloadable performance reports that can be viewed in third-party gradebook programs or Microsoft® Excel. Students can always review the results of any previously completed assignments.

For more information, see *Working with the Gradebook (page 49)*.

1.4 System Overview

Table 1.1: System Modules Overview

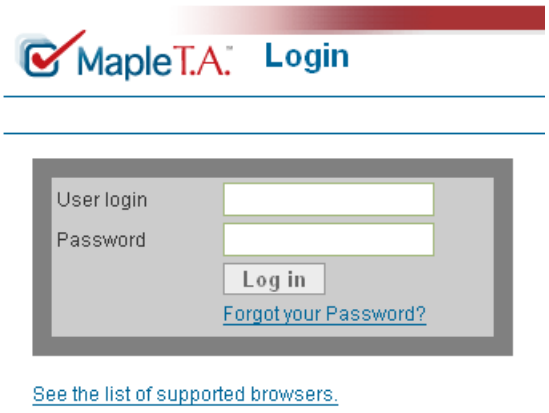
Assignment Editor	• Create and publish assignments • Set rules and policies
Question Repository	• Create questions to use in your tests, quizzes, and assignments • Store and manage questions
Gradebook	• View individual and class results • Export to an external Gradebook • Force grade incomplete assignments
Class Manager	• Edit your class information • Lock registration for your class • Share your class content with others
User Manager	• Register users into your class • Remove users from your class • Upload a class roster • Export a class roster
Additional Tools	• Change your password • Proctor tools • Upload and manage reference files used in questions and assignments • Course modules

See Also:

Security Issues to Consider (page 69)

The System Login

All users log in to the system using a username and password.



Login Screen

For instructors, the system administrator provides your username and password. The administrator or instructor provides students with a username and password to log in.

Maple T.A. can be configured to allow self-registration. If this feature is enabled, users will see a link on the login page allowing them to register themselves. Ask Maple T.A. system administrator at your university about your configuration.

System Homepage

Logging into the system brings you to the **System Homepage**. If you have created any classes, they are listed under the **Classes I Am Instructing** section (see **Figure 1.1**).

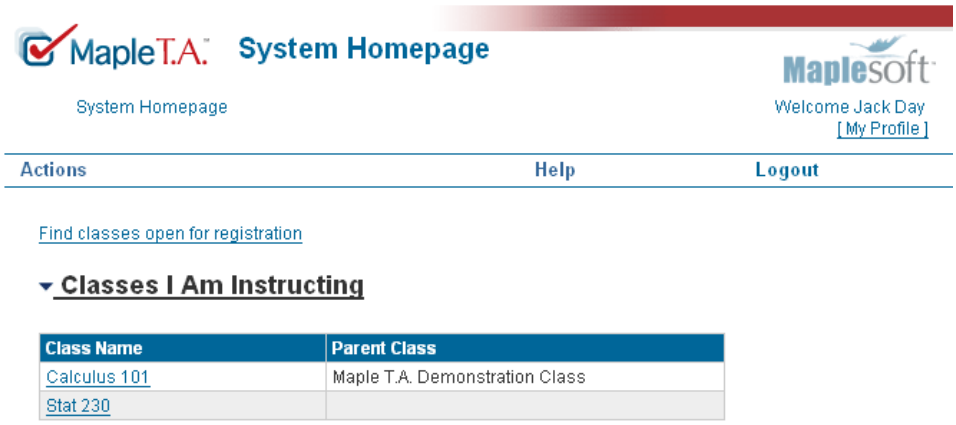


Figure 1.1: System Homepage with Classes I Am Instructing

Additionally, any classes in which you are a student or proctor are also listed, under their own sections (**Classes I Am Taking**, and **Classes I Am Proctoring**, respectively).

From the System Homepage, click on a class name link to open the **Class Homepage**.

The Class Homepage

Once you have created your class, you can upload a student roster or allow students to register themselves in your class. The **Class Homepage** is used by the instructor, students, and proctors involved in that particular class.

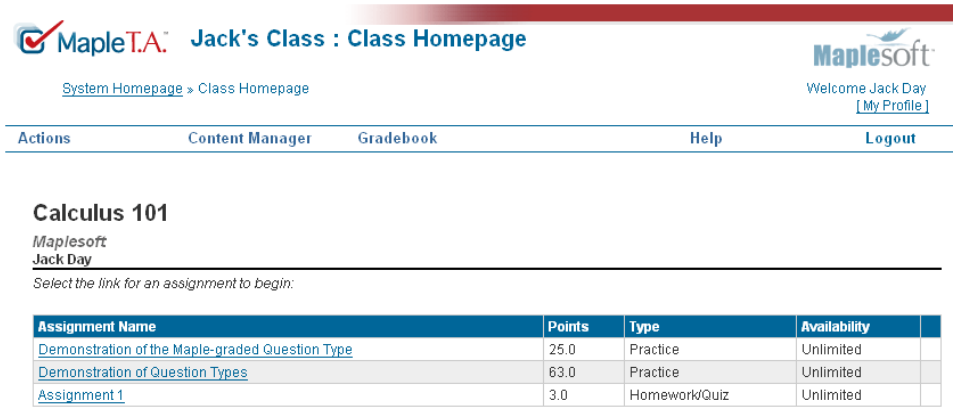


Figure 1.2: Class Homepage with Assignment List

See Also:

Creating a New Class (page 7)

Using the System in the Classroom

Possible Uses of Maple T.A. for Online Homework and Testing

- Create self-scoring, web-delivered tests and homework assignments from your time-tested problem files.
- Use formatted problem sets to create automated, self-grading homework or test assignments.
- Edit, modify, or expand existing questions.
- Control the rules and policies surrounding your assignments, including time limitations, ability to retake assignments, feedback content, and results reporting.
- Publish assignments to your class-specific web site.
- Use the system **Gradebook** to monitor individual and class learning performance, and export data to leading course management systems like WebCT™, Blackboard®, and Moodle™. Content and student records in the system are built in XML and MathML. The system output is IMS QTI (IMS Question & Test Interoperability specification) compliant, improving interoperability with other systems and enabling you to attach additional information to your content.

Getting Help

From the **Help** menu, select **Instructor**. You can access Proctor and Student help from the **Other Help** submenu.

Access Technical Support from Maplesoft

- Email your questions to **support@maplesoft.com**

Maplesoft

615 Kumpf Drive

Waterloo, Ontario

CANADA

N2V 1K8

2 Creating and Managing Classes

2.1 Class Manager

Creating a New Class

You begin using Maple T.A. by creating a class. Within your class, you have access to the question repository and assignments, resource files, and the **Gradebook** for the class. It also gives you access to several system tools.

Note: Your administrator may have already created a class for you. If this is the case, once you login you can access your class by clicking the link under the **Classes I Am Instructing** tab.

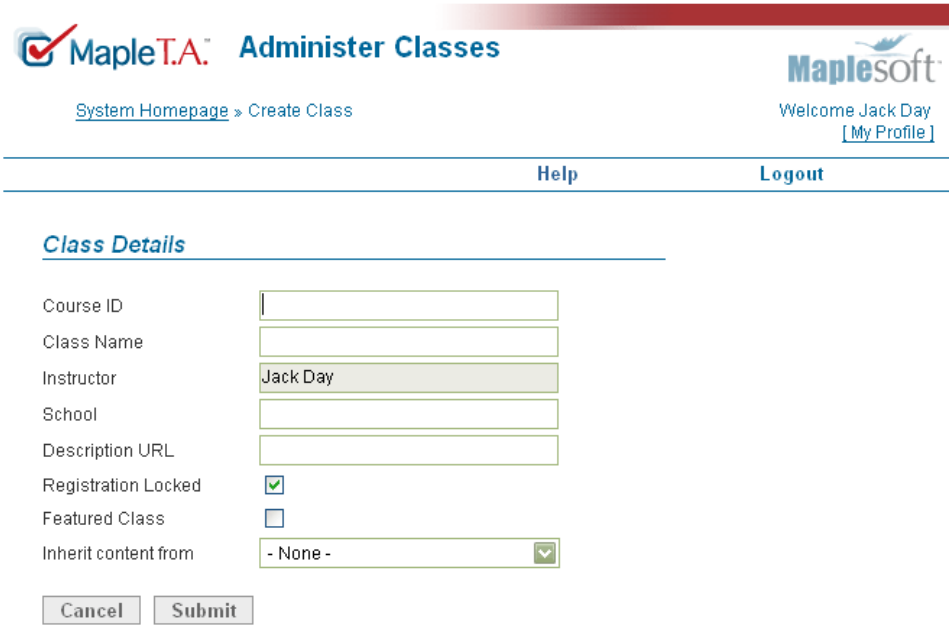
Use one of the following options to create your new class:

Create a New, Empty Class

Note that you can only create a class if you have been given the correct privilege by the Administrator. If you cannot create a class, contact your Administrator.

To create a new class in the system:

1. From the System Homepage, select the **Actions** menu, and then select **Add Class**.
2. Complete the class registration form, and click **Submit**.



Create a New Class

3. Your class is added to the list of **Classes I Am Instructing**.
4. To access your class, click the link to your newly created class to bring you to the **Class Homepage**.

Create a Copy of a Shared Class

To create a copy of a shared class (to create that class that includes questions and pre-populated assignments from the parent class):

1. From the System Homepage, select the **Actions** menu, and then select **Add Class**.
2. Complete the class registration form, and select a class to **Inherit content from**. See **Figure 2.1**.
3. Click **Submit**.

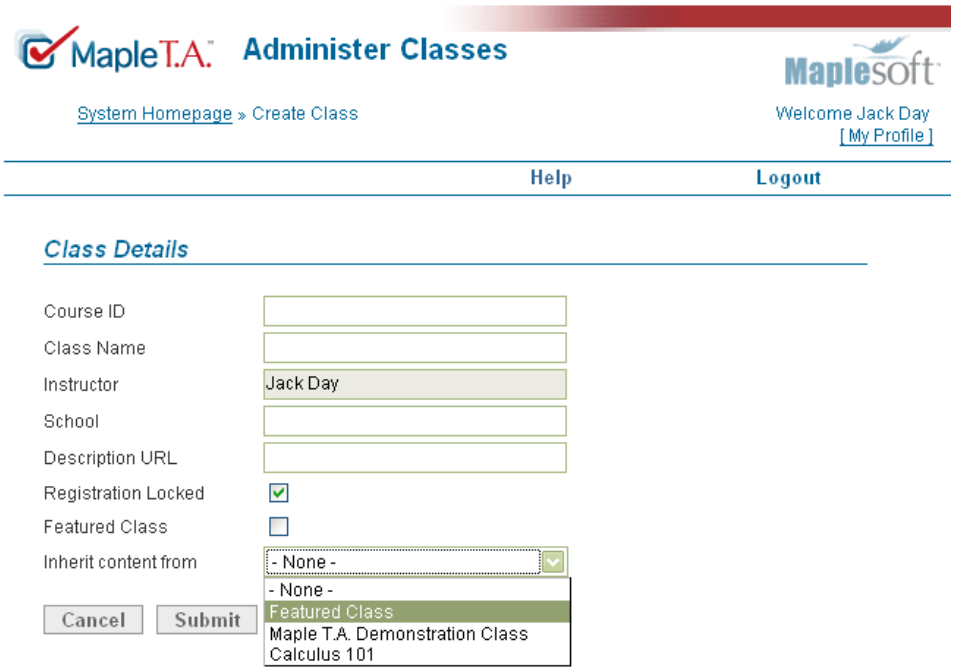


Figure 2.1: Inherit Content from a Featured Class

3. Your class is added to the list of **Classes I Am Instructing**.
4. To access the newly created class, click the link to the class to bring you to the **Class Homepage**.

The classes that appear on the list that you can inherit content from are called *featured classes*.

See Also:

Shared Classes Overview (page 13)

Create a Child Class

In addition to inheriting content from a featured class, you can create a copy of any class for which you are the instructor.

To create a child class (to create that class that includes questions and pre-populated assignments from the parent class):

1. Go to the Class Homepage for the class you want to share.
2. From the Class Homepage, select the **Actions** menu, and then select **Add Child Class**.
3. If desired, click **Change Instructor** and select a new instructor from the system users. Select the radio button beside the user and click **Update** to change the instructor.
4. Complete the class registration form, and click **Submit**.
5. The class is added to the list of **Classes I Am Instructing** (if you are the instructor).
6. To access the newly created class, click the link to the class to bring you to the **Class Homepage**.

Required Registration Information

The following information is requested when you register a new class in the system:

Course ID

(optional) A combination of letters and numbers to identify the course; for example, *stats101*.

The maximum number of characters allowed is 10.

Class Name

The displayed name for a class.

The class name appears on the Class Homepage and in other administrator lists. The **Class Name** is created at the time of initial registration by the instructor or the system administrator. An administrator can rename classes.

- The maximum number of characters allowed is 255.
- The system ensures that class names are unique. If you or an instructor select a name that exists, the system requests that you select another name.
- Clicking the **Class Name** link opens the Class Homepage in the related class.

Instructor

The instructor's name.

An instructor is selected from the database of system users at the time of initial class registration and setup. The user selected to be the instructor does not need to have the default system role of Instructor; any user in the system can be made an instructor. This is useful because a student user can be made the instructor of a tutorial class, for example. Then the student will have student privileges in those classes in which he or she is enrolled as a student, and he or she will have instructor privileges in the class that he or she is instructing.

School

(optional) The name of your school.

Description URL

(optional) You can provide the URL of a web page with additional information about the class.

Registration Locked

Registration is locked by default, preventing students from enrolling themselves in the class. Open registration allows students in the system to enroll themselves in the class.

Featured Class

Marking a class as a featured class allows for the creation of child classes that inherit content from this class.

Inherit Content From

You can inherit content from a featured class. This gives you access to questions, assignments, quizzes, and tests from the featured class.

Registering Users in a Class

You can register users in a class by:

- Allowing students to register themselves for your course
- Creating and uploading a class roster
- Using the User Manager

If you allow self-registration, students can register for you class using the **Find classes open for registration** link on the **System Homepage**.

Class rosters are an efficient way to add a large number of students to your course at once. The user manager can be used to search for and add selected students individually.

To register students in your course, they must have an account in the system. Generally, your system administrator creates user accounts. The system administrator can give instructors the ability to create user accounts. Contact your system administrator for more information.

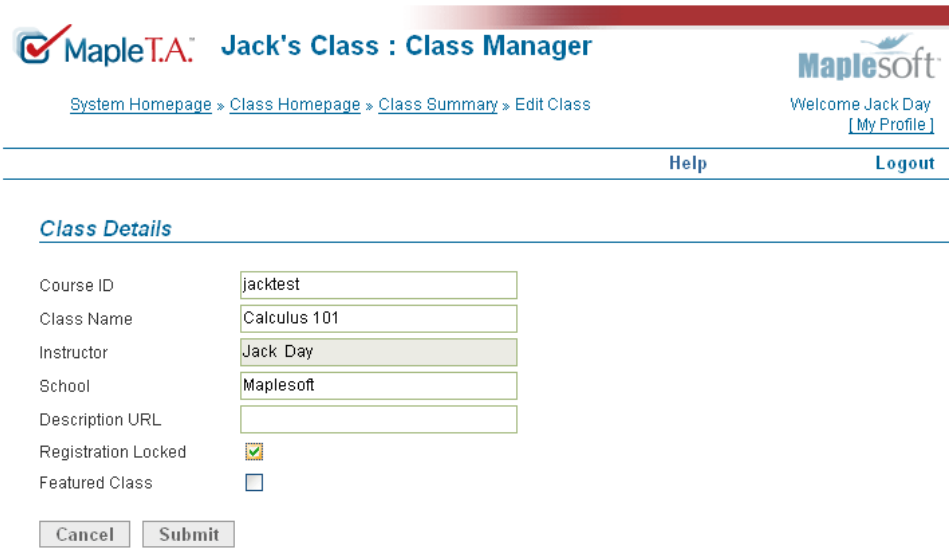
Locking and Unlocking Class Registration

You can close registration for your class to prevent students from registering for your class.

To lock class registration:

1. From the Class Homepage, select the **Actions** menu, and then select **Class Info**.
2. Click the **Edit Class** button.
3. Select the **Registration Locked** check box. (This is selected by default.) See **Figure 2.1**.
4. Click **Submit**.

To allow open registration for the class, you can leave the class unlocked for the first two weeks of class and then lock it. If you do not want to allow open registration, you can lock the class from the start.



Locking Class Registration

Class Rosters

You can use class rosters to:

- Upload a class roster to the system from a text file. If a user is not in the system, and you have *create privileges*, the student will be added to the system and registered in the class.
- Update the Maple T.A. database with any new information in the roster by uploading a new class roster for an existing class.
- Remove students from a class by deleting a roster.

To view your user privileges, click the **My Profile** link on the top right of the screen.

To create and upload a class roster:

1. From the Class Homepage, select the **Actions** menu, and then select **User Manager**.
2. From the **Actions** menu, select **Roster>Import**.
3. Click **Browse** to find the text file containing the class roster. The text file (for example, ***.txt** and ***.rst)** can contain fields delimited by comma or tab characters. You can upload an Excel spreadsheet which has been saved to a text file. The file must have:
    - One line for each student
    - A header for each included column that matches the specified headers.

Note: If your university uses LDAP authentication, the usernames in the import file should be their LDAP usernames. In this case, the passwords will be the LDAP passwords. See *LDAP Authentication (page 71)*

4. Click **Submit**. The records are displayed. If there are any errors, these are displayed.
5. Click **Enroll Roster**. The students are added to the system, if appropriate, AND registered in your class. Print this list to record login and password information, or click **Create CSV** to save this information to a file. **Important:** If this page is not printed, automatically generated passwords will be lost.

Attributes:

- **Login** must be at least one alphanumeric character and must be unique.

- **First Name** must be at least one alphabetic character.
- **Middle Initial** must be no more than one alphabetic character.
- **Last Name** must be at least one alphabetic character.
- **Email** must be a unique address.
- **Student ID** must be at least one alphanumeric characters and must be unique.
- **Password** must be at least five alphanumeric characters.
- **Role** can be one of student, guest, proctor, instructor, or administrator. If roles are specified, they cannot exceed the level of the current user. The default if not specified is student.
- **Require User Validation** can be yes or no. If yes, the next time the user logs in to Maple T.A., they will be asked to validate their profile. The default is yes.

The corresponding header row values are:

login, first, initial, last, email, student, password, role, validate

The headers are not case-sensitive, so *first* and *First* are both recognized.

The other fields are optional. If no password is provided, the system generates a password, which is displayed on the next page.

If you have a null field (for example, a student did not provide a middle initial), use two consecutive delimiters. However, if the header row includes email and student ID, you cannot have empty values for those fields in the rows of student data. If there are some students who do not have either an email or a student ID, you should use the value <NULL>.

To upload an updated roster for a class:

You can update profile information through roster uploads. Follow the same steps as for creating and importing a roster. To update a roster, only the login username field is required.

To remove students from a class:

From the Class Homepage, select the **Actions** menu, and then select **User Manager**. From the **Actions** menu, select **Roster>Delete**. Select the file containing the users you want to remove from your class roster.

To export a roster:

Click on Export options: **CSV** to open or save the class roster to a CSV (comma-delimited) file

The User Manager

You can select students to register in your class from a list of system users. All the students you want to register must already have an account in the system. You can search for users that satisfy particular criteria.

To register users using search criteria:

6. From the **Class Homepage**, select the **Actions** menu, and then select **User Manager**. From the **Actions** menu, select **Register Users**.
7. Enter search criteria to narrow the list of users down or to search for a particular student.
8. Select the check boxes beside the students you want to register in your class.
9. Click **Register**.

Shared Classes

You can share your questions, resources, and assignments with other instructors. Class sharing is an ideal way to manage multiple class sections of a large course using common questions and assignments, ranging from practice sessions to exams. Shared classes can also contain unique questions and assignments that are not part of the parent class.

When you share a class, you create a single parent class whose content and assignments are cloned. The content and assignments are *inherited* by each child class when a new class is created and the **Inherit content from** option is selected in the class registration form.

Use class sharing to administer a large course with many individual class sections. With class sharing, you can draw assignments from a common pool of questions, administer common homework, practice assignments, or exams. Even though content and assignments can be managed in common, each individual section can contain unique content and assignments.

Questions and assignments inherited from a parent class are always identified in lists by the shared content icon (), which appears beside the assignments and questions.

Shared Class Behaviors

- Changes, additions, and deletions made to questions and assignment content in the parent class are automatically distributed to child classes.
- The content of an inherited assignment cannot be edited directly. However, you can clone an inherited question to make your own copy, which you can edit.
- If you edit the policies of an inherited assignment, your changes are reflected in the assignment sessions taken by students in your class immediately after you save your changes.

Important: In general, changes to questions or assignment content in the parent class **automatically update** those questions or assignments in your child class. This is not true about polices; changes to assignment policies are not inherited. You control the assignment policies for an assignment that you inherit from a parent class.

See Also:

Sharing Questions (page 117)

Shared Class Content Updates

The following applies to changes you make to inherited questions, whether you make changes to them individually using the **Question Editor** or if you use course modules to make global changes that affect inherited questions.

Editing Inherited Content or Installing Replacement Content with the Same Name

If you install a course module that has a new version of a question or assignment with the same name as that in the parent class, **you break the link** to the parent class version of the question or assignment. The same rule applies if you edit a parent question or assignment and then save/install it using the original name of the parent class.

If you want to refresh the content of your class to match that in the parent class or restore the link to the parent class version of the content, you must save your new, edited question or assignment under a different name, or delete the edited/modified version of the question or assignment. The parent class restores the matching (original) question/assignment automatically.

Implications

If you modify questions or assignments that you inherit from a shared parent class, there is no risk that the parent class version will overwrite your modified version, unless you specifically delete the modified question or assignment from your child class. This means that if you make changes to the inherited questions or assignments and save them to the

original name, your changed versions sever the relationship with the parent asset and you will not receive any future updates from the parent class automatically. Therefore, you must stay in communication with the instructor in your shared parent class if you want to be notified of changes to parent class questions and assignments.

See Also:

Course Modules (page 61)

Importing and Installing a Course Module (page 64)

Creating a Shared Class

To share the contents of your class with other instructors:

1. From the **Class Homepage,** select the **Actions** menu, and then select **Class Info**.
2. Click the **Edit** button.
3. Select the **Featured Class** check box.
4. Click **Submit** to save your changes.

Other instructors can create a copy of your class, including your questions and assignments, by creating a new class and choosing your class from the **Inherit content from** list.

To remove class sharing:

1. From the **Class Homepage**, select the **Actions** menu, and then select **Class Info**.
2. Click the **Edit** button.
3. Clear the **Featured Class** check box.
4. Click **Submit** to save your changes.

See Also:

Creating a New Class (page 7)

2.2 Manage Proctors

You can add a proctor to your class to ensure security in exam settings. Proctors can log in to their own computer and perform proctoring duties using the Proctor Tools utility. Proctors can authorize students to start and grade a test. To authorize students, the proctor must verify the student's identity and then enter his or her password to allow the student to begin or to grade the test. A proctor can enter his or her password at the student's workstation (local authorization), or authorize the student from his or her own computer (remote authorization). The benefit of using remote authorization is that the proctor does not need to enter his or her password in front of students.

When a student attempts a test that is proctored, an **Authorize** screen is displayed. This screen indicates that the student requires authorization. Typically, a proctor is physically present in the same computer lab during testing sessions. For convenience, when notified by the student, proctors can authorize a student exam session from their own work station using the **Proctor Tools** screen, and then inform the student. See **Figure 2.2**.

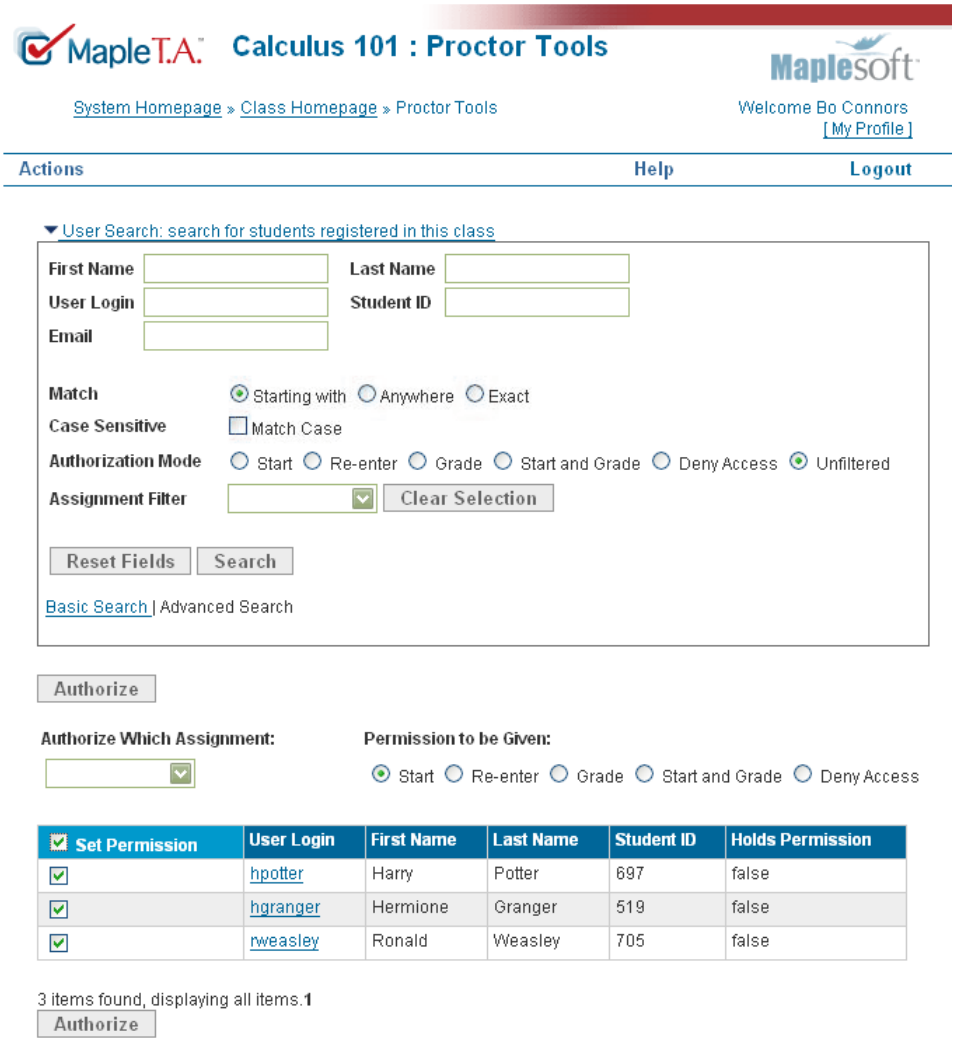


Figure 2.2: Proctor Tools

For a list of proctors in your class, select the **Actions** menu, and then select **User Manager**. In the Search panel, select the **Proctor User Role** radio button, and click **Search**. A list of proctors is displayed.

See Also:

Proctored Exams (page 35)

Adding a Proctor

To add a proctor to your class:

1. From the Class Homepage, select the **Actions** menu, and then select **User Manager**.
2. From the **Actions** menu, select **Register Users**.
3. Search for, or select, a user from the list to be registered as a proctor in your class.
4. Select the **Proctor** User Role radio button.
5. Click **Register**.
6. Now that user will be able to access your Class Homepage and use the Proctor Tools utility.

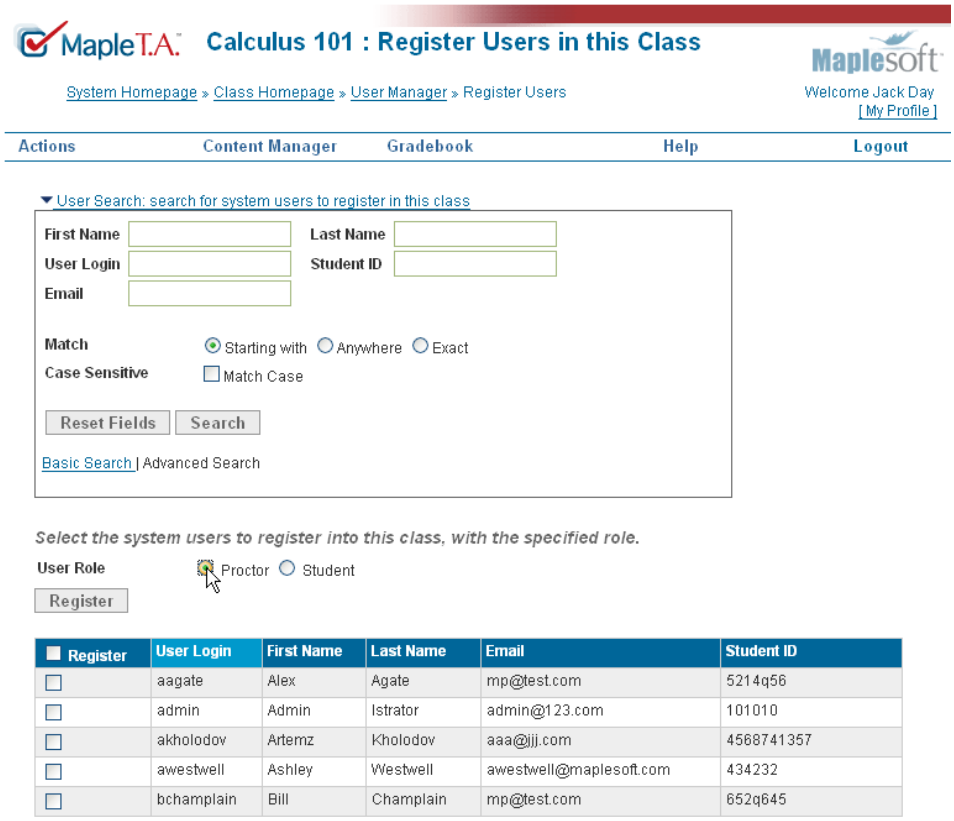


Figure 2.3: Add a Proctor to Your Class

Note: A user with the role of **Proctor** (set by an administrator, or an instructor with additional privileges) has proctor privileges in any class in which he or she is not registered as a student. This means that such a user does not need to be registered into the class to perform proctor duties in your class.

Deleting a Proctor

To delete a proctor:

1. From the **Class Homepage**, select the **Actions** menu, and then select **User Manager**.
2. From the **Actions** menu, select **Remove Users**.
3. In the Search panel, select the **Proctor** radio button, and click **Search**. A list of all proctors for the class are displayed.
4. Select the check box next to the user you want to remove, and click **Remove**.

2.3 The Class Web Site Editor

In conjunction with your class homepage, your class also has a web site folder on the system that you can use to store images and other resource material used in your questions. The **Web Site Editor** provides a list of the contents of the web site and gives you options for uploading, organizing, and deleting files.

In the class **Web Site Editor**, you can perform these actions:

- *Uploading Single Files (page 18)*
- *Uploading Multiple Files (page 18)*
- *Working with Folders in the Class Web Site (page 19)*

You can access the **Web Site Editor** from the **Class Homepage**. Click **Content Manager**, and then select **Web Site Editor**.

Figure 2.4: Web Site Editor

Note: Your class web site is also accessible directly from the **Question Editor** when you author a **Clickable Image** question.

Uploading Single Files

To upload a single file:

1. From the **Web Site Editor**, navigate to an appropriate subfolder on the web site or click the **Create new subfolder** icon to create a subfolder. For maintainability, it is recommended that you place files in a subfolder.
2. Click the **Upload files(s) to this point...** icon.
3. Click **Browse** to locate and select the file. The following file formats are supported: htm, html, gif, jpg, png, pdf, xls, class, jar, js, css, shtml, exe, zip, swf, lib.
4. Select the radio button next to **Single File**. The system uploads the file to your class web site, giving it the name that you specify. (The default name is the name of the file on your computer.)
5. Click **OK** to perform the upload and return to the list of files on your web site. The uploaded zip file appears with the list of files in the web site explorer.

Important: The maximum size of a file that can be uploaded is 5 MB.

Uploading Multiple Files

You can upload multiple resource files to your class web site if you package them in a compressed **zip** file first. The **Zip Archive** option is useful if you have many files to upload.

To upload multiple files:

1. From the **Web Site Editor**, navigate to an appropriate subfolder on the web site or click the **Create new subfolder** icon to create a subfolder. For maintainability, it is recommended that you place files in a subfolder.
2. Click the **Upload files(s) to this point...** icon.
3. Click **Browse** to locate and select the zip file.
4. Select **Zip Archive**. The program extracts all of the files from the selected **zip** file and uploads them to the web site. (If the selected file is not a **zip** file, the program does not upload any file.)
5. Click **OK** to perform the upload and return to the list of files on your web site. The uploaded zip file appears with the list of files in the web site explorer.

Important: The maximum size of a zip file that can be uploaded is 5 MB.

Viewing or Deleting Files

To view or delete a file:

1. Click the file icon next to the name of the file. The full URL of the file is displayed.
2. To verify the contents of the file, click the URL to open the file in a separate browser window. Close the browser window once you are sure that you have the right file.
3. Click **Delete** to delete the file.
4. A confirmation message is displayed before this file is deleted.

Working with Folders in the Class Web Site

Creating Subfolders

To create a subfolder:

1. Click the folder icon next to **new subfolder**.
2. Enter the name of the subfolder. When creating the name, use letters, numbers, or underscores in the name, but *not* spaces.
3. Click **OK**.
4. The program adds the subfolder at the *top* of the web site folder and opens it. If the web site has more than one subfolder, the program arranges them in alphabetical order. Each subfolder has its own icons for creating a new subfolder and for uploading files. Use these to create sub-subfolders and to upload files to the subfolder.

Closing Subfolders

When a subfolder is open, the web site explorer does not show the subfolder or upload icons for the root level of the web site (the top level).

To close a subfolder:

1. Click the triangle icon by the name of the subfolder. You return to the root level.
2. Click the triangle icon again to open the subfolder and work on its files.

Deleting Subfolders

To delete a subfolder:

1. Click the triangle icon by the name of the subfolder to open the subfolder.
2. Click **Delete**.
3. A confirmation message is displayed. Click **OK** to delete the subfolder.

Note: If you click **Delete** when you are at the root level (that is, when no subfolder is open), an error message appears indicating that you cannot delete the **Public_Html** directory, which is the main folder for the Class Web Site.

3 Creating and Managing Assignments

3.1 The Assignment Editor

Assignments are created by selecting questions from the Question Repository. The **Assignment Editor** allows you to create new assignments, edit the content, properties, or appearance of existing assignments, and copy, print, or delete assignments. Assignment creation options enable you to create assignments in which Maple T.A. reorders questions, generates questions, or displays a subset of questions. Each student viewing your assignment can potentially complete a unique set of questions.

The Assignment Editor lists assignments created for your class. If you began your class registration by selecting a template or a shared class (not an *empty class*), it may already be populated with a series of assignments based on testing materials, or other shared questions. If not, you will see a blank list here initially.

To access the Assignment Editor:

- From the Class Homepage, select the **Content Manager** menu and then select **Assignments**.

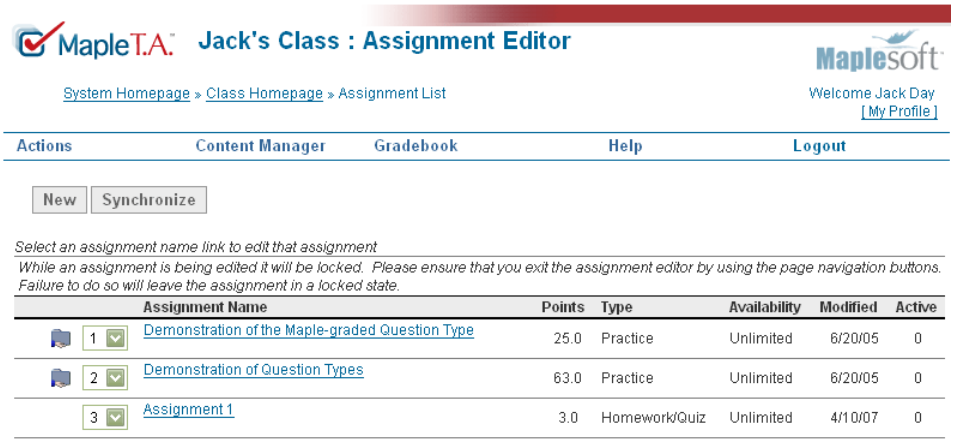


Figure 3.1: Assignment Editor

Overview: The Assignment Editor Screen

Options in the Assignment Editor main window	Uses
New	Allows you to create new assignments. (*Adding New Assignments (page 23)*)
Synchronize	Allows you to refresh any inherited assignments.
Assignment Name	Allows you to open and modify existing assignments. (*Editing Assignments (page 25)*)
Edit	Allows you to edit existing assignments. (*Editing Assignments (page 25)*)
Copy	Allows you to copy existing assignments. (*Copying Assignments (page 23)*)

Options in the Assignment Editor main window	Uses
Delete	Allows you to erase existing assignments. (*Deleting Assignments (page 24)*)
Hide/Show	Allows you to hide the assignment from student view in the Class Homepage. (*Assignment Visibility (page 46)*)
Print	Generates a version of your test that is suitable for printing and distributing to students as a pencil-and-paper assignment. (*Printing Assignments (page 26)*)
Summary	Allows you to view a summary of the assignment properties.
Force Grade	Allows you to force grading of an active assignment. (*Forced Grading (page 59)*)
Drop-down Arrow	Use to sequence your assignment list. (*Controlling the Sequence of Listed Assignments (page 26)*)
Shared Content Icon	Indicates the assignment is an inherited assignment. Appears next to an assignment in the list when the assignment originates in the *parent* class from which your class was created. (*Adding New Assignments (page 23)*) See also *Shared Classes Overview (page 13)*
Locked Assignment Icon	Indicates the assignment is locked because there is an active version of the assignment (a student has started but not finished the assignment).

Warnings and Locking Mechanism

When an assignment is opened for edit, the system checks whether the assignment has associated student records in the Gradebook or is currently in use by a student. The number of students currently using the assignment is displayed in the **Active** field of the assignment list in the **Assignment Editor**.

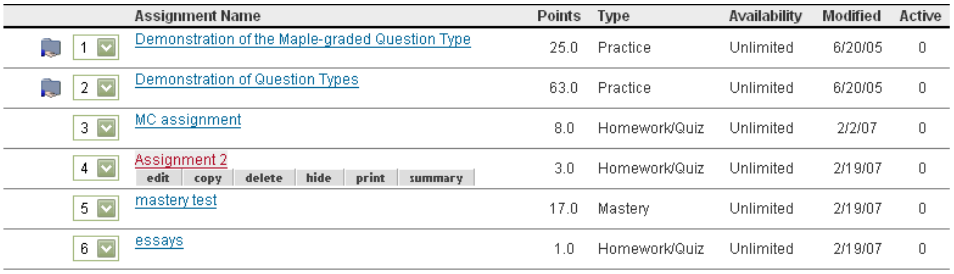
- If the assignment is currently in use, the Locked Assignment icon is displayed beside the assignment name. The questions in the assignment are locked so that you cannot edit them. You can edit some of the policies of the assignment, such as the time limit.
  It is possible to force grade incomplete assignments so that you can edit the assignment. However, if a student is actively in the process of taking an assignment, you cannot force grade the student's assignment.
- If the assignment is not currently active, the assignment is *locked for editing* so that new active tests cannot be started during the editing process. Students who try to start the assignment while it is locked are notified with a warning message.

See Also:

Assignment Types Overview (page 33)

Overview: Using the Assignment Editor (page 26)

Options in the Assignment Editor



	Assignment Name	Points	Type	Availability	Modified	Active
1	Demonstration of the Maple-graded Question Type	25.0	Practice	Unlimited	6/20/05	0
2	Demonstration of Question Types	63.0	Practice	Unlimited	6/20/05	0
3	MC assignment	8.0	Homework/Quiz	Unlimited	2/2/07	0
4	Assignment 2 edit copy delete hide print summary	3.0	Homework/Quiz	Unlimited	2/19/07	0
5	mastery test	17.0	Mastery	Unlimited	2/19/07	0
6	essays	1.0	Homework/Quiz	Unlimited	2/19/07	0

Figure 3.2: Options in the Assignment Editor

From the list of assignments in the Assignment Editor, you can modify and manage existing assignments.

To view the options, hover your mouse over the assignment name. The option buttons are displayed.

- Edit assignments by clicking **edit**.
- Copy assignments by clicking **copy**.
- Delete assignments by clicking **delete**. (The delete option is not available for shared assignments.)
- Hide assignments from student view in the **Class Homepage** by clicking **hide**.
- Print assignments. Clicking **print** displays the print preview of the assignment. You then have the option to **Print** the assignment or click the **Back** button to return to the **Assignment Editor**.
- View a summary of the assignment by clicking **summary**.
- Force grade a locked assignment by clicking **force grade**. (The force grade option is only available for locked assignments.)

Adding New Assignments

To add a new assignment:

1. In the **Assignment Editor,** click the **New** button to create a new assignment. The **Assignment Editor** screen is displayed with the **Choose Name** tab active.
2. Follow the instructions in *Overview: Using the Assignment Editor (page 26)*.

Copying Assignments

To copy an assignment:

1. In the **Assignment Editor,** move the mouse over the assignment (name) to be copied. Six option buttons are displayed. (See **Figure 3.2**.)
2. Select **copy**.
3. "Copy of <original-assignment-name>" appears in **Assignment Name** list.
4. Click this copied assignment link.
5. Enter a new name in the **Choose a name for your assignment** field.
6. Make changes as necessary to the question set (**Select Questions** tab) and policies (**Set Policies** tab).
7. Click the **Review & Finish** tab, and then **Finish** to save the assignment with the new name (and updates).

Deleting Assignments

Considerations BEFORE you delete an assignment:

- After scheduled assignment deadlines have passed, the assignment is listed on the student's view of the assignment menu for your class, but cannot be selected.
- Hiding an assignment or scheduling an assignment for a specific time period may allow you to achieve most of your objectives for deleting an assignment.

To delete an assignment:

- In the **Assignment Editor,** move the mouse over the assignment (name) to be deleted. Six option buttons are displayed. Click **delete**, and a confirmation prompt appears.

  If you click **Yes**, the program removes the assignment from the list in the **Assignment Editor** screen and from lists of assignments elsewhere in the system. Students will not be able to select the assignment from the list of assignments in the Class Homepage and you will not be able to view the students' past results for the assignment in the **Gradebook**.

Important: Deleting an assignment causes you to lose all results for that assignment from the **Gradebook**.

Disabling an Assignment

To *disable* an assignment so that students cannot see it, you can make it hidden. Clear the **Visible** check box under Assignment Properties in the **Set Policies** tab of the **Assignment Editor**.

If an assignment is flagged as *hidden*, the Class Homepage does not include it in the list of assignments offered to students. However, a student can view past results for the assignment in the gradebook. For more information on hiding an assignment, see *Assignment Visibility (page 46)*.

Alternatively, you might schedule the assignment to be available for a specific time period. Before and after the indicated availability window, the assignment is shown in the assignment list on the Class Homepage but cannot be selected.

Deleting Shared Assignments

The system does *not* allow you to delete an inherited assignment that appears in your assignment list as part of a shared class or a template.

To disable a shared assignment:

- In the **Set Policies** tab of the **Assignment Editor**, clear the **Visible** check box under Assignment Properties. For more information, see *Assignment Visibility (page 46)*.

Restoring a Hidden Assignment

To make an assignment available:

1. Return to the **Set Policies** tab of the **Assignment Editor**.
2. Change the **Visibility** of the assignment to **Visible**, or reset the scheduled availability so that it appears again on the Class Homepage.

Editing Assignments

To edit an existing assignment:

- Click the assignment you want to edit. The **Assignment Editor** menu opens with the **Choose Name** tab active.

Operations within the **Assignment Editor** are identical whether you are creating a new assignment or editing properties of an existing assignment. All tabs are accessible.

From within the **Assignment Editor** you can:

- Change the name of the assignment
- Add or delete questions
- Group, organize, or sequence question delivery
- Attach or edit policies

Finalizing Changes

To finalize your changes:

1. Select the **Review & Finish** tab.
2. Verify your edits.
3. Click **Finish**. You are returned to the **Assignment Editor**, listing all the assignments for your class.

Tip: From within the **Assignment Editor** you can move between tabs in any order. The numbers on the tabs are merely a suggested sequence.

See Also:

Finishing and Saving an Assignment (page 47)

Editing Shared Assignments

You cannot make changes to inherited assignments in the same way as you edit any assignment in your class. You can control some assignment policies for inherited assignments, however you cannot delete or modify them.

To modify an inherited assignment, create a copy of the assignment.

When you copy a shared assignment, your copy of the assignment automatically appears under its new name in the **Assignment List**. Note that the original assignment—inherited from the shared or template class—also appears in the list. If you do not want the original assignment to appear, remove it from view by clearing the **Visible** check box under **Assignment Properties** in the **Set Policies** tab of the **Assignment Editor**.

Important: It is possible to break the link to a parent assignment by uploading a course module that has a question with the same name. For more information, see *Importing and Installing a Course Module (page 64)*

Previewing Assignments

You can preview an assignment from the student's perspective.

1. From your **Class Homepage**, click on an assignment. The assignment opens in a new window.
2. Your results are recorded in the Gradebook, but the data is not included in statistical reports (since you are an instructor).

Previewing Questions

To preview individual questions within an assignment, use the **Assignment Editor**. See *Previewing Questions in Assignments (page 40)*.

Printing Assignments

To print an assignment:

1. In the **Assignment Editor**, move the mouse over the assignment (name) to be printed. Six option buttons are displayed.
2. Select **print.** The system displays the list of questions in the browser and opens a second browser window that contains a **Back** button and a **Print** button.
3. Review the assignment.
4. Click the **Print** button in the second browser window to print the assignment, or click the **Back** button to discard the assignment and return to the **Assignment Editor**.

Controlling the Sequence of Listed Assignments

The list of assignments that is displayed in the **Assignment Editor** is also displayed to students in your Class Homepage. You can change the order of assignments listed in the **Assignment Editor**. These changes are reflected in the order of assignments in your Class Homepage.

To order the list of assignments:

- In the **Assignment Editor**, select the new position number from the drop-down menu of the assignment you want to reposition. Once selected, all affected assignments will adjust their position accordingly.

Overview: Using the Assignment Editor

The **Assignment Editor** organizes assignment creation into a four step process.

- Naming your assignment
- Selecting questions
- Establishing rules and policies
- Reviewing, finishing, and publishing assignments to your class

To create a new assignment:

1. From the **Content Manager** menu, select **Assignments**. The **Assignment Editor** opens. (See **Figure 3.1**.)
2. Click the **New** button.
3. The **Assignment Editor** screen opens with the **Choose Name** tab active.
4. Follow steps 1 through 4. More advanced options are described in detail later in this chapter.

Step 1: Select a Name for the Assignment

Begin by giving the assignment a name.

Step 2: Select Questions for the Assignment

1. View and add questions to your assignment.
2. Set question weighting.
3. Scramble question sequence upon delivery.

Step 3: Set Assignment Policies

1. Select the type of assignment and set assignment options.
3. Control feedback upon grading.
4. Set assignment properties.
5. Control assignment scheduling.
6. Create rules between assignments.

Step 4: Review and Finish Assignment

1. Review assignment content, rules, and policies.
2. Save the assignment by clicking **Finish**.

Important: The **Assignment Editor** does not save partially completed assignments until you click **Finish** in the **Review & Finish** tab. If you begin to create an assignment and then abandon it without clicking **Finish**, your additions and changes will be lost.

Assigning Partial Credit

You can administer partial credit within assignments for partial answers, alternate answers, or alternate forms of answers. Maple-graded Questions and Multipart Questions offer you flexible ways to assign partial credit.

Multipart questions allow you to create problems with many steps and to grade intermediate responses at the completion of each step. Assigning partial credit in multipart questions is closely related to establishing point values for the individual (simple) question parts during construction of the multipart (complex) question in the **Question Editor**.

By carefully structuring and considering the weighting of your questions, you can control partial credit in multipart questions very specifically. You must set the value of each component part at the time of question creation. See *Multipart Questions in the Question Editor (page 100)*.

Maple-graded Questions allow you to specify partial credit for certain answers. See *Maple-graded Questions in the Question Editor (page 96)*.

Many related approaches offer instructional advantages:

- Using Question Groups within assignments
- Controlling Question Weighting within assignments
- Using Mastery Mode assignments

3.2 Tutorial: Using the Assignment Editor

Assignment Editor Step 1: Choose a Name for the Assignment

To name your assignment:

1. In the **Choose Name** tab, enter an appropriate assignment name for your assignment in the **Choose A Name For Your Assignment** field. The assignment name is displayed to students on the Class Homepage.
2. Optional. To add assignment-level page headers and text for the results page, click the **Advanced** button.

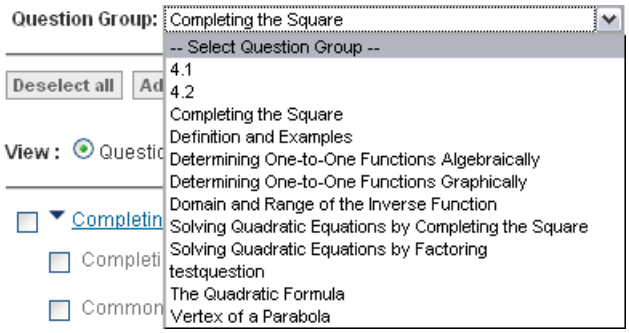
Proceed to Step 2: Select Questions.

Assignment Editor Step 2: Select Questions

In the **Select Questions** tab of the **Assignment Editor**, you can:

- Add questions individually
- Select by search criteria
- Select groups of items and add them all at once, either as individual items or in pools or groups of questions

1. To add questions to an assignment, select a source question group from the **Question Group** drop-down list.



Select Question Group

2. Expand the groups and select the questions you want using the check boxes on the left. Preview the question if necessary.
3. Add individual questions:
   - Click the **add** link to the right of the question.
   - Select several questions using the check boxes and click the **Add as Items** button
   - Click the check boxes beside the topic name to select every question in the topic and click the **Add as Items** button.

Individual questions are included in every version of the assignment served to a student and you control question weighing individually.

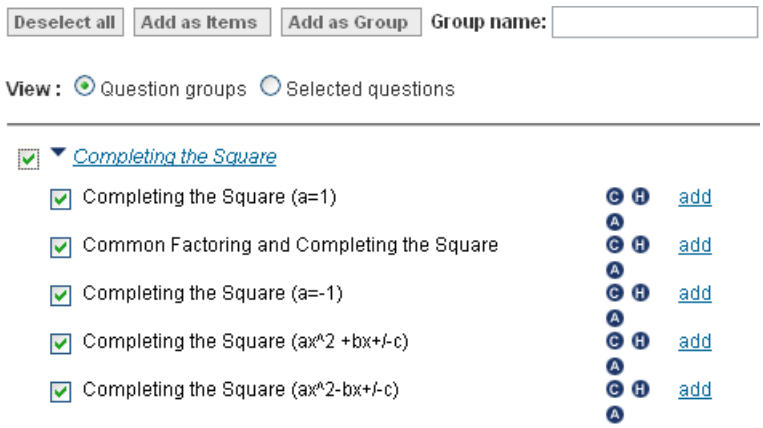


Figure 3.3: Add Questions to Assignment

4. Add a group of questions:

    a. Select questions using the check boxes.

    b. Enter a name for the **Group name** and click the **Add as Group** button.

Adding questions in question groups provides you with the added option of having the system select from within the questions at random according to the criteria you specify (for example, choosing three of seven questions from the group each time a new assignment is created for a student). When assigning question weighting for a question group, you set a single point value that controls the weighting of all questions drawn from the group. See **Figure 3.4**.

5. Optional.

    - Change the default question weighting (one point per question) to match your requirements.
    - Select the check box to scramble the sequence of question delivery when assignments are created.
    - Reorder questions within your assignment.
    - Merge individual selected questions to form new question groups in your assignment.
    - Include annotations along with individual questions in your assignment.

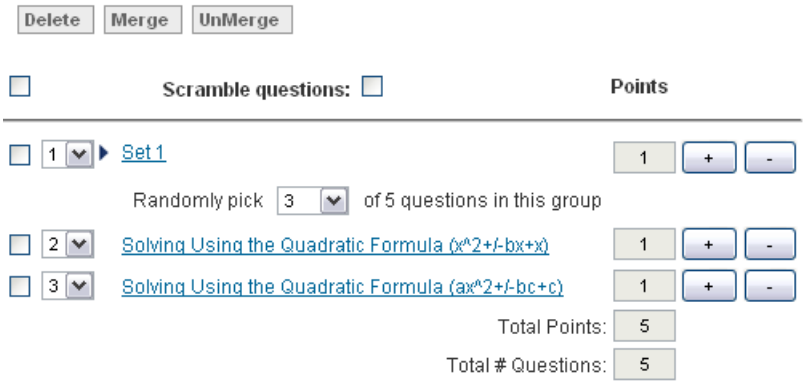


Figure 3.4: Select Questions: Specify Order and Weighting of Questions

When you have completed these steps, proceed to Step 3: Setting Policies.

Assignment Editor Step 3: Setting Policies

In **Set Policies** tab of the **Assignment Editor** screen, you can:

- Select the type of assignment
- Decide when to make the assignment available to your class
- Set feedback options for the assignment
- Establish other grading policies

Type of Assignment

Assignments for Credit (Results are Recorded)

There are three assignment types that record students results. This is an assessment style session that records performance in the class **Gradebook** after students click **Grade**. For recorded assignments, students can return to view their assignment results (with exactly the same algorithmic data values) in the future.

- *Homework or Quiz Assignments (page 34)* is the default assignment type
- *Proctored Exams (page 35)*
- *Mastery Assignments (page 36)*

Assignment Options

- *Rework the Same Assignment Option (page 34)*
- *Proctored Exams (page 35)*
- *Setting Mastery Policies (page 43)*

Practice Assignment Types (Results are not Recorded)

There are two assignment types that allow students to practice. For practice assignments, graded results are not recorded in the **Gradebook**. Students cannot return to view their results later.

- *Anonymous Practice Assignments (page 35)*
- *Study Session Assignments (page 37)*

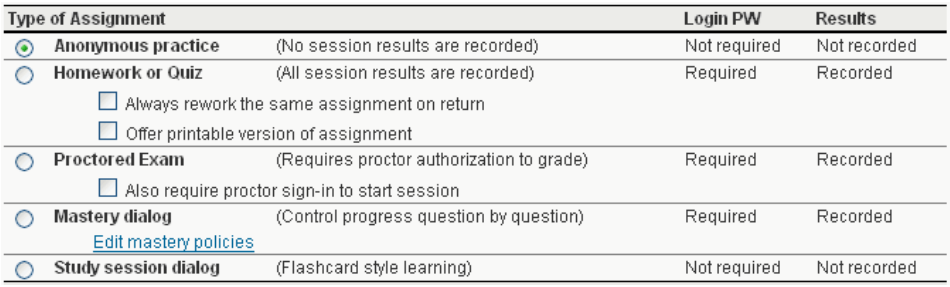


Figure 3.5: Setting the Assignment Type

Feedback

During the assignment, you can:

- Provide hints
- Show the question comment (if they exist)
- Allow students to check grades and receive feedback

After the assignment is graded, **Show the final grade** and **show the correct answer** are default settings.

Other options for after the assignment include:

- Receiving email reports whenever a student completes an assignment of type Homework/Quiz or Proctored Exam. The email will contain the class name, assignment, student name, and their grade.
- Showing the question comment (if it exists)
- Restricting feedback until a specified date

Restricting feedback until the assignment due date is useful if you want to prevent students who finish the assignment early from giving answers to other students.

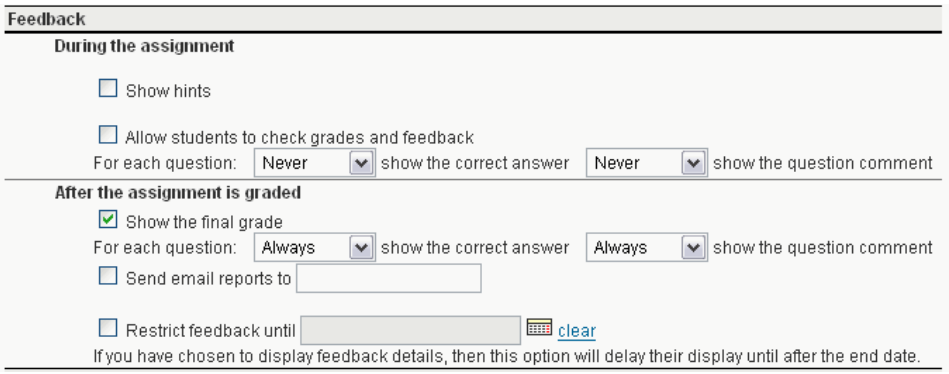

Feedback
During the assignment
Show hints
Allow students to check grades and feedback
For each question: Never show the correct answer Never show the question comment
After the assignment is graded
Show the final grade
For each question: Always show the correct answer Always show the question comment
Send email reports to
Restrict feedback until clear
If you have chosen to display feedback details, then this option will delay their display until after the end date.

Figure 3.6: Feedback Options

Optional: Set Assignment Properties

You can set a passing score, time limit, the number of questions on a page for the assignment, and the maximum number of attempts allowed.

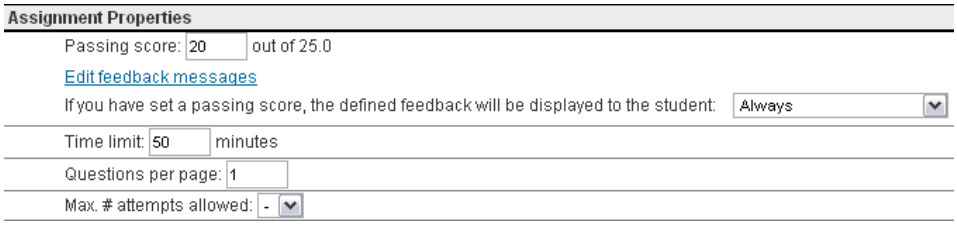

Assignment Properties
Passing score: 20 out of 25.0
Edit feedback messages
If you have set a passing score, the defined feedback will be displayed to the student: Always
Time limit: 50 minutes
Questions per page: 1
Max. # attempts allowed: -

Figure 3.7: Assignment Properties

Control Assignment Scheduling

You can set a default start and end date and time for each assignment. The default is no scheduled start and end date.

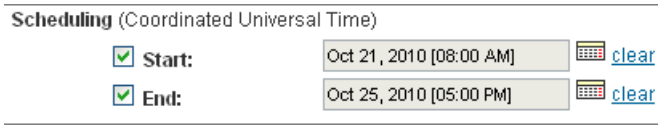


Figure 3.8: Scheduling

Visibility of Assignment

Clearing the **Visible** check box temporarily excludes the assignment from the list of assignments displayed to students on your class homepage.

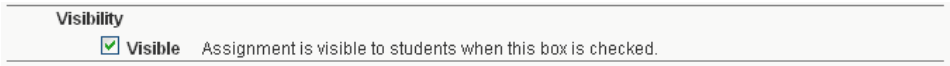


Figure 3.9: Visibility

Optional: Create Rules between Assignments

You can restrict access to an assignment so that only students meeting certain criteria can take the assignment.

Set criteria by clicking the **Advanced** button.

When you have completed your assignment rules and policies, go to Step 4 :Review & Finish.

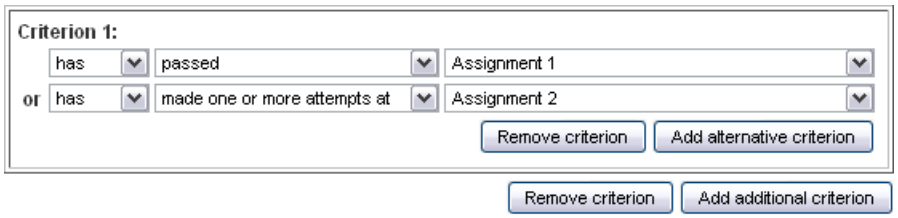


Figure 3.10: Set Criteria for Assignments

Assignment Editor Step 4: Review & Finish

In the **Review & Finish** tab of the **Assignment Editor**, you can review assignment summary information, including:

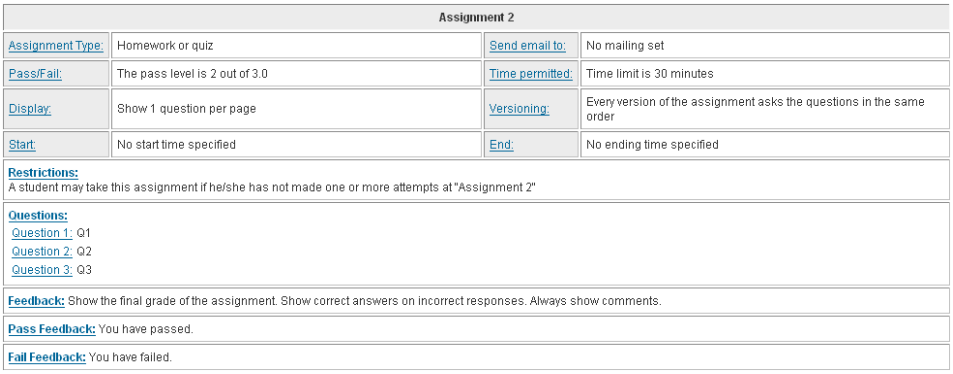
- Assignment Type
- Questions Selected
- Scheduled Availability
- Rules and policies you established

Previewing Questions

To preview a question, click the related link while in the Assignment Editor.

Editing Options

- If you want to make a change, clicking the link for each category returns you to the appropriate **Assignment Editor** tab.
- If your assignment is ready to save, click **Finish** to return to the **Assignment Editor**, where you will view your list of assignments, with the new assignment displayed in the list. Your edits are immediately saved to your class records. If you set up your assignment to be immediately available to students, it appears in the list presented to them in the Class Homepage.



Assignment 2			
Assignment Type:	Homework or quiz	Send email to:	No mailing set
Pass/Fail:	The pass level is 2 out of 3.0	Time permitted:	Time limit is 30 minutes
Display:	Show 1 question per page	Versioning:	Every version of the assignment asks the questions in the same order
Start:	No start time specified	End:	No ending time specified
Restrictions: A student may take this assignment if he/she has not made one or more attempts at "Assignment 2"			
Questions: Question 1: Q1 Question 2: Q2 Question 3: Q3			
Feedback: Show the final grade of the assignment. Show correct answers on incorrect responses. Always show comments.			
Pass Feedback: You have passed.			
Fail Feedback: You have failed.			

Figure 3.11: Review Assignment Details

See Also:

Setting Assignment Types (page 42)

Scheduling Assignments (page 45)

3.3 Assignment Types

Assignment Types Overview

You can create graded assessments (homework, quizzes, or exams), ungraded practice tests or quizzes, or tutorial assignments with set criteria. For details, see:

- *Homework or Quiz Assignments (page 34)*
- *Proctored Exams (page 35)*
- *Anonymous Practice Assignments (page 35)*
- *Study Session and Mastery Assignments (page 36)*

To take any assignment, students must be registered in the class.

You can configure the number of questions on a page, create test instruction sheets and headers, and set policies for restricted access, repeated attempts, and other options.

- Students answer questions one-at-a-time, and the system records their responses between questions.
- Jumping from question to question within an assignment is allowed until students complete all questions (or choose to ignore the warnings to do so) and submit their responses for automatic grading.
- For **Homework/Quiz** and **Proctored Exam** assignments, individual student results are recorded in the **Gradebook**.

For a summary of assignment attributes, see **Table 3.1**.

Table 3.1: Assignment Attributes

Assignment Type	Graded	Recorded in Gradebook	Hints Available	Solution Available
Anonymous Practice	yes	no	yes	no
Homework/Quiz	yes	yes	yes	no
Mastery	yes	yes	yes	no

Assignment Type	Graded	Recorded in Gradebook	Hints Available	Solution Available
Proctored Exams	yes	yes	yes	no
Study Session	yes	no	yes	yes

Homework or Quiz Assignments

Description

- Students are presented with an assignment consisting of any number of instructor-selected questions.
- Questions are delivered either in an instructor-specified or scrambled sequence.
- Student responses during sessions are recorded (but not graded) after every question, so assignment sessions can be interrupted and returned to upon next login.
- By default, homework or quiz assignments can be attempted multiple times for credit. If you want to limit the number of attempts a student can make on a particular assignment, configure the assignment properties.
- Results for every attempt at a **Homework and Quiz** assignment are recorded in the Gradebook for the class.

Rework the Same Assignment Option

To present an exact version of an assignment each attempt:

Select the **Always work the same assignment on return** check box.

- Individual students will always return to the exact version of the assignment they were initially presented with, even on repeated attempts, regardless of whether the question was authored with algorithms for any attempt during the assignment's scheduled availability. If questions contain randomly generated variable data, the system instantiates the variables exactly one time.

To vary questions for each assignment attempt (default setting) :

Leave the **Always work the same assignment on return** option cleared.

- Students rework the assignment with newly randomized variables inserted in questions each time they attempt the assignment. This enables instructors to create a *template* assignment and lets students rework it repeatedly, with each session generating a different version of the algorithmic variables within the assignment, but still covering the same skills.

Notes: The system records graded results for every repeated attempt, and instructors can control the grading policies, taking the best score, average score, or another option. You can view results from all sessions, including the unique randomized variable data for each assignment attempt a student makes. The system records the student's best grade. The system still produces algorithmically varied versions of the assignment for different students, but each student sees only one version of the assignment regardless of repeated attempts.

Printable Assignments Option

To make your assignments printable:

Select the **Offer printable version of assignment** option.

- With this option selected, students login and the system generates a new assignment (with algorithmically generated variable data where appropriate). At the next step, however, students are offered the option to generate a printable version of the test suitable for printing and working offline.
- Printable assignments operate like a homework session that was interrupted before grading. The student can log in again and enter responses for automatic system grading and recording scores in the Gradebook. Or, instructors may prefer to have students submit their completed assignments on paper for hand-grading by instructors or teaching assistants.

Associated Assignment Properties

You can set the passing score and time limit for an assignment. These properties are set under Assignment Properties in the **Set Policies** tab.

- If there is a time limit set for the test in the assignment definition, a pop-up window warns students as they approach the time limit. They are not allowed to answer additional questions once they exceed the time limit; however, they can submit their work up to that point for grading.
- When the student has completed the assignment and submitted it for grading, the system displays a graded assignment report that includes the score (for example, 6 out of 8) as well as details on the grading of each question, along with comments or feedback for all of the questions. (These options can be configured.)

Proctored Exams

Proctored Exams are similar to **Homework** or **Quiz** assignments, but with an additional security measure to confirm the identity of the student taking the test.

- All Proctored tests require a proctor to authorize students submitting their test for grading.
- You can require proctor authorization to validate student identity and grant assignment access at the start of a proctored exam. Select the **Also require proctor sign-in to start session** option in the **Set Policies** tab.
- Students must be registered in your class.

Types of Proctors

There are two types of proctors: **Global** and **Local**.

- Global proctors are defined by the *System Administrator* and can give authorization for any class.
- Local proctors are defined by the *Instructor* and can only give authorization for a particular class.

Location of Proctor

Proctors can give authorization directly at the student's computer, or remotely through **Proctor Tools**. In both cases the proctor must sign in by giving a login name and password.

Note: To give authorization remotely, the proctor must login and access the **Proctor Tools** utility under the **Actions** menu.

See Also:

Adding a Proctor (page 16)

Managing Proctors (page 14)

Anonymous Practice Assignments

Anonymous Practice assignments are similar to Homework or Quiz assignments; however, results are not recorded in the **Gradebook**. Any number of questions can be included, and the assignment is delivered in an instructor-specified sequence (or scrambled).

Graded assignment reports are produced upon conclusion of the assignment, but results are not recorded in the **Gradebook**. Students can view their results at the end of the session; no permanent record is maintained. Results can be printed, but are only available for as long as a student displays them.

Study Session and Mastery Assignments

Study Session and Mastery assignments usually draw from a large pool of assignment material (often algorithmically generated to produce limitless question permutations). There is no pre-established number of questions for either Mastery or Study Session assignments, questions are drawn from pools throughout the study session.

Assignments are delivered one-question-at-a-time, and the student submits each question for grading individually and immediately, instead of having the entire test graded at the end. These assignments are also sometimes described as *learning dialogues*, because of their question-answer-next question style.

Mastery and Study Session assignment types emphasize the following pedagogical approaches:

- For **Mastery** assignments, instructors create carefully structured collections of questions grouped by learning objectives. These assignments usually draw from a large pool of assignment material, often algorithmically generated to produce limitless question permutations. Mastery assignments require a login and password, and results are recorded in the **Gradebook**.
- For **Study Sessions**, students control the study process by practicing question after question. Questions can provide hints and full solutions that are available to students while they work. Results are displayed one question at a time and are *not* recorded in the **Gradebook**.

Mastery Assignments

For **Mastery** assignment types you first create an assignment by selecting questions (Step 2) as with other for-credit assignments. However, when selecting questions for a mastery assignment you must give careful pedagogical consideration to structuring questions by learning objective.

Question groups can be especially important in the instructional design of **Mastery** assignments. In Step 3 - Setting Policies, you create rules that establish the minimum number of questions in each learning objective that must be answered correctly before the student can move to the next learning objective.

For students, the graded results of their responses on individual questions at each level of learning objective determines their progression through the assignment. When students demonstrate the required level of mastery (by answering the minimum number of questions correctly), the system allows them to move ahead to the next learning objective's material.

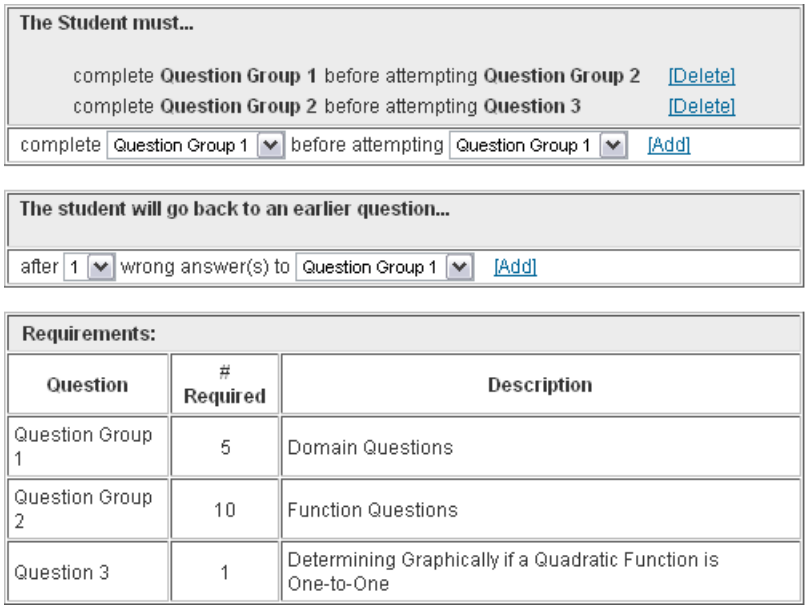

The Student must...
complete Question Group 1 before attempting Question Group 2 [Delete]
complete Question Group 2 before attempting Question 3 [Delete]
complete Question Group 1 before attempting Question Group 1 [Add]
The student will go back to an earlier question...
after 1 wrong answer(s) to Question Group 1 [Add]
Requirements:
Question
Required
Description
Question Group 1
5
Domain Questions
Question Group 2
10
Function Questions
Question 3
1
Determining Graphically if a Quadratic Function is One-to-One

Figure 3.12: Policies for Mastery Assignments

If students demonstrate a lower level of mastery (through repeated incorrect responses), you can establish penalties so that the system moves the student backwards, to the previous learning objective until minimal proficiency is again established.

Requirements

- Assignment results are recorded in the **Gradebook** for the class.
- Mastery assignments can be attempted multiple times for credit. If you want to limit the number of attempts a student can make on a particular assignment, configure the assignment properties.
- As students work through a **Mastery** assignment, they are shown information about their progress along with the requirements for the learning objective and the assignment. This information is shown in a **Progress Report** box that updates each time they complete and grade a question.

Study Session Assignments

Study Sessions draw from a bank of questions and serve repeated practice to students one question at a time. These assignments are principally a teaching tool, whereas other types of assignments might be considered testing tools (although all assignment types do both to some degree). This assignment type is ideal for anonymous, student-driven, self-paced review.

Although you can create a specific content structure for study session assignments, you can also allow the system to randomly select questions from a pool of questions. Consequently, **Study Session** assignments typically draw from Question Groups. Students work at their own pace on questions drawn from a pool of instructor-selected questions.

Where hints and solutions are part of the question content, students can use them during Study Sessions.

Grading

Questions are graded one at a time, and results are not recorded in the **Gradebook**. When the system grades the question, it does not tell students the correct answer, but simply whether they got the question correct. If they answered correctly, it proceeds to the next question. If they answered incorrectly, they can view the **Hints** and then return to make another attempt at the question, or they can view the solution and proceed to the next question. Students can also skip questions and leave the test at any time.

3.4 Details on Using the Assignment Editor

Choose Name Tab

Assignment Name

The name you give your assignment in the **Choose Name** tab of the **Assignment Editor** is displayed to students in your Class Homepage.

Name assignments using unique and informative terms that make it easy for students to identify each assignment and its instructional objective. For example, although the rules for grading and access are enforced automatically, you may want to indicate the type of assignment in the name for rapid identification.

Assignment Instructions

This feature is useful for providing instructions viewed by students during an assignment.

1. In the **Choose Name** tab, click **Advanced.**
2. In the first text box, enter text that will appear at the top of every page in your assignment.
3. You can also enter information to be shown on the results page. Details are in the section **Text Instructions Shown Upon Grading**.

Text Instructions Shown Upon Grading

To include text or instructions:

1. In the **Choose Name** tab, click **Advanced.**
2. Enter text or instructions in the **Text to show on the Results Page** field. This information is displayed to students after grading, as part of the graded assignment report they receive. The editor allows you to format your text, as well as include images, links, and color enhancements.

See Also:

Configuring Feedback Options (page 43)

Question Annotations (page 40)

Select Questions Tab

Add Questions to Assignments

You can add questions to your assignment by

- Adding individual questions by using the **Add as Items** button

or

- Adding questions as a group by using the **Add as Group** button

You can use any combination of selection methods when creating an assignment.

You can combine any number of individual questions and question groups in a single assignment.

Adding Individual Questions

To add individual questions:

1. In the **Assignment Editor**, **Select Questions** tab, select a source question bank from the **Question Group** drop-down list.
2. Expand the groups into individual questions.
3. From the left frame of the **Select Questions** tab, select questions using the check boxes.
4. Click the **Add as Items** button to place the selected questions in your assignment.

Note: You can select one question at a time and click the adjacent **add** link to place the individual question in your assignment.

See Also:

Question Annotations (page 40)

Add Questions as a Group

Questions added as a group are loaded into a single question pool. You can use the drop-down menu to specify the number of questions to be selected by the system from this pool during your assignment. Question groups are useful in designing a Mastery assignment.

To add questions as a group:

1. In the **Assignment Editor**, **Select Questions** tab, select a source test bank from the **Question Group** drop-down list.
2. Expand the groups into individual questions.
3. From the left frame of the **Select Questions** tab, select questions using the check boxes.
4. Enter a name in the **Group name** box. (Note: For a Mastery assignment, these names are shown to the student in the progress report shown after each question is graded.)
5. Click the **Add as Group** button to add the questions into your assignment as a group or question pool.

See Also:

Mastery Assignments (page 36)

Expanding Groups to View Individual Questions

After selecting a source question bank, the groups for that bank appear in the *left* frame of the **Select Questions** tab of the **Assignment Editor** screen.

To display questions in a group:

1. Click the ▸ icon to expand the group. When expanded, the icon changes to ▾.
2. To close the group and its question contents, click ▾.

Delete Questions from Assignments

To delete questions that have been selected for inclusion in an assignment:

1. In the *right* frame of the **Select Questions** tab, select questions using the check boxes to the left of the question name.
2. Click **Delete**.

Merge Questions into Question Groups

You can merge individual questions or question groups into a single question group in an assignment.

To merge questions:

1. In the *right* frame (displaying the selected questions), select the questions or question groups you want to merge.
2. Click the **Merge** button to combine all the selected questions into a single question group.

Previewing Questions for Selection

When you select a question group, its structure is displayed in the left frame, directly below the question group list.

To view brief descriptions of individual questions, expand the groups and view the question description. If no question description accompanies the question, its question type is displayed (for example, **Multiple Choice**, **Formula**, and **Fill-in-the-Blanks**).

To preview individual questions for selection:

- In the left frame, click the link of the question you want to preview. A preview window opens. You can also preview a question after it is added to the assignment by selecting the question link in the right frame.

Previewing Questions in Assignments

After questions have been added to an assignment, you can preview the questions.

To preview a question in an assignment:

1. In the *right* frame of **Select Questions** tab, select a question.
2. Click the link in the highlighted question description. The question preview includes the information fields and the Annotation Editor.

Question Annotations

Question Annotations (in the form of HTML or unformatted text) can be added to individual questions in assignments. Annotations can be used to provide:

- Additional instructions
- Explanations before grading
- Links to outside materials or reference objects
- Informational text between questions.

Creating an Assignment Annotation

1. In the *right* frame of the **Select Questions** tab, select a question and click its associated link to launch the preview window.
2. Scroll down in the question preview window to find the **Annotation Editor** (shown below).

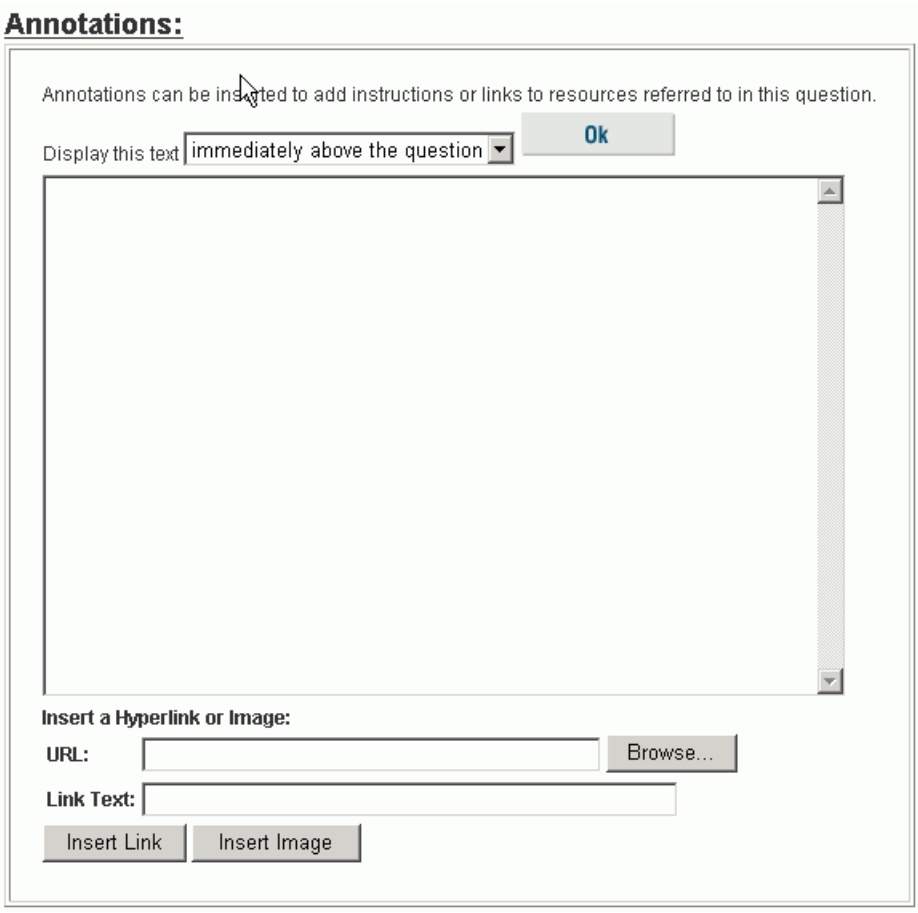


Figure 3.13: Assignment Annotations

3. Enter annotation text (including HTML formatting) in the large text entry field.
4. Set options for displaying your annotation. From the **Display this text** drop-down list, select from:
    - immediately above the question
    - immediately below the question
    - in the header of the page
    - in the footer of the page

Use the link editor at the bottom of the window to identify reference file sources, either image files or web references, from your Class Web Site or outside web addresses. URLs must be of the form: http://www.website.com/referencefile.ext

Enter any **Link Text** you want displayed identifying the reference object or web address.

See Also:

Assignment Instructions (page 38)

Text Instructions Shown Upon Grading (page 38)

Configuring Question Weighting

By default, each new question in your assignment is added with a weight of 1 point per question. To change the weighting of individual questions, change the point value in the **Points** column for the question. The total points at the bottom of the column automatically adjust as you change individual question weights.

Note: The question score values you assign here apply only to Assessment-style assignments such as, Homework or Quiz, Anonymous Practice, and Proctored Exam. Mastery and Study Session Assignments do not use these settings because they employ other pedagogical scoring models.

Scrambling Question Delivery

To automatically scramble the sequence of questions delivered each time a new assignment is presented to a student, select the **Scramble Questions** option within the **Select Questions** tab.

Note: In the **Review & Finish** tab, question scrambling is referred to as **Versioning**. This term is used because scrambling the sequence of question delivery produces multiple *versions* of assignments. Using algorithmically generated questions within assignments creates multiple versions of questions as well.

Set Policies Tab

Setting Assignment Types

Homework or Quiz is the system default assignment type. It delivers an assessment-style session that does not require proctor authorization. It records student session results in the class **Gradebook**.

To change the assignment mode, select one of the other options under **Type of Assignment** in the **Set Policies** tab of the **Assignment Editor**.

For information on each assignment type, see:

- *Anonymous Practice Assignments (page 35)*
- *Homework or Quiz Assignments (page 34)*
- *Proctored Exams (page 35)*
- *Mastery Assignments (page 36)*
- *Study Session Assignments (page 37)*

Note: If you select **Mastery dialog** assignment, you should proceed to the Mastery Policies page to specify the criteria for your assignment.

Setting Mastery Policies

If you select the **Mastery** type for the Assignment, you can set learning objectives or competency levels that must be met before a student is allowed to proceed with the assignment.

To establish policies for your assignment:

1. In the **Set Policies** tab of the **Assignment Editor**, click the **Edit mastery policies** link below the **Mastery dialog** option. The **Set Policies/Mastery Policies** screen opens. See **Figure 3.12**.
2. Use the drop-down fields to establish rules. The field text provides guidelines.

Penalties apply to the selected topic and also to any preceding (or *controlling*) topics on which the selected topic depends. The system returns students to the earlier topics where a learning need has been demonstrated until they reach your required level of competency.

Example

Consider an assignment in which Topic B depends on Topic A. Both have a basic requirement of 3 correct questions, and Topic B has a penalty of one additional question for every wrong answer. If the student answers three questions correctly in Topic A and then moves to Topic B but gets a question wrong, both topics now have a requirement of 4 correct questions, so the student no longer meets the requirements for Topic A. The system returns to Topic A and the student must answer an additional question correctly in order to proceed again to Topic B.

Note*:* When delivering **Mastery** assignments, the system gives students information about their progress towards meeting the requirements for the assignment. It updates this information with every question that it grades.

See Also:

Setting Assignment Requirements: Limiting Student Access to Assignments (page 46)

Configuring Feedback Options

You can specify which feedback elements are displayed to students when they grade an assignment. (See **Figure 3.6**.)

The default system behavior is to show the final assignment grade as well as question feedback, but you can switch either of these elements on or off, control access to hints during assignment sessions (before grading), or withhold feedback until after the assignment due date has passed.

Hints and Feedback

You can allow access to question hints (if they exist) in any assignment type. **Hints** and **Question Feedback** (contained in the **Comment** field of a question) are not always present in the source question bank content. You can add hints or comments by using the **Question Editor** and editing these fields for individual questions. If you choose to display hints and comments and your source questions lack them, the system ignores the display settings but otherwise provides a grading report to students.

In a grading report, the system displays question feedback from the **Comment** field if it exists, or if not, shows the correct answer value (as defined in the **Answer** field). While every question will have something displayed as *feedback* in the grading report, not every question may have the type of extensive feedback, coaching, or external review references that you might provide in the comment for a question.

Note: To display hints during an assignment or custom feedback after grading, your questions **must** contain content in **hint** fields or the **comment** field.

Selecting Feedback Options

To display hints during an assignment, select the **Show hints** check box.

To display a grade and/or feedback for each question while the student is taking the assignment, select the **Allow students to check grades and feedback** check box.

To display the final grade after the assignment is graded, select the **Show the final grade** check box.

You can show feedback during the assignment or after it is graded, for example, displaying the correct answer or comment for each question. Under **Feedback**, select from **Always**, **If correct**, **If incorrect**, or **Never** condition options for displaying the correct answer and question comment.

You can edit the grading message and feedback messages, if you would like to use custom messages instead of the default messages.

Receiving Email Notification

You can opt to receive an email whenever a student completes the assignment in **Homework** and **Proctored Exam** mode, giving the name of the student and their grade. To receive email notification, select the **Send email reports to** check box and enter your email address.

Restricting Feedback

You can restrict feedback so that feedback details are not displayed until a certain date. Check the **Restrict Feedback** check box and click the calendar icon to choose a date.

See Also:

Security Issues to Consider (page 69)

Configuring Assignment Properties

Assignment properties are configured in the **Set Policies** tab of the **Assignment Editor.** These include:

- *Setting a Passing Score (page 44)*
- *Setting a time limit (page 45)*
- *Setting questions per page (page 45)*
- *Setting the Maximum Number of Attempts (page 45)*

Setting a Passing Score

If you select the **Practice**, **Homework**, or **Proctored Exam** assignment types, you can set a **Passing score** for the assignment. If you set a score, the system assesses each try at the assignment as either **Pass** or **Fail**, and records this information in the Gradebook automatically. The field shows the total available score for the assignment (for example, *out of 10*), which varies according to the composition of your assignment. For Homework and Proctored Exams, if you choose to set a passing grade and you have selected to receive email notifications when students complete the assignment, the emails will included information on whether the student meets the passing criteria.

You can customize the message students receive by clicking the **Edit feedback messages** link. You can use the drop-down menu to specify when the feedback is displayed: **Always**, **Never**, or **If the final grade is shown**.

Setting a Time Limit

You can set a time limit for an assignment or test. If you set a limit, the program shows the student the time remaining during the course of the test. If the time limit expires during the test, the system informs the student, and does not allow the student to answer any more questions. However, the student can submit the test for grading.

Note: The timer does not stop until the student runs out of time or clicks **Grade**. The timer will continue to run even if the student clicks **Quit and Save**.

Setting the Number of Questions per Page

By default, the program displays one question per page when presenting an assignment to students. You can use this option to deliver more than one question per page. If you set the number of questions to be greater than 5, a warning dialog will appear. If you have used question annotations and set the annotations to display at the top or the bottom of the page in which the question is loaded, your annotations appear as a page-level header or footer (rather than immediately before or after the question).

Note: Mastery and Study Session assignments automatically serve one question at a time. This is part of their instructional design and this setting cannot be changed for these assignment types.

Tip: If you are concerned about students' ability to access complicated questions over a slower online connection, it is recommended that you accept the default setting for questions per page (which is one-at-a-time). Loading one question at a time usually allows a student to move through the assignment effectively, but also saves each question response as it is completed.

Setting the Maximum Number of Attempts

You can set the maximum number of times a student may take an assignment. If the student attempts more than the maximum amount, a message is displayed indicating the restriction.

It is possible to allow an exception in a particular case to the maximum number of attempts restriction. To authorize an exception, when the student sees the restriction message, the student clicks **Exception**. A proctor or the instructor can then provide authorization.

Scheduling Assignments

Using the calendar function under **Scheduling** in the **Set Policies** tab, you can set **Start** and **End** times for each assignment. These times govern the availability of the assignment to students in your class. Scheduled times refer to your system's time zone setting (indicated in the Scheduling section, and set by your System Administrator).

Before and after the indicated availability window, the assignment is displayed in the assignment list viewable by students on your Class Homepage; however, it cannot be selected. Note that for the instructor, unavailable assignments appear in the list of assignments on the Assignment Editor page. Additionally, unavailable assignments will appear in a student's past results page if the student completed them.

Notes:

- You can also choose to withhold grades and feedback until after the assignment due date by setting these policies under **Feedback** in the **Set Policies** tab in the **Assignment Editor**.
- Students must complete their assignment sessions by submitting their work to be graded before the time and date specified in the assignment schedule.
- You can also set time limits for assignment sessions. If there is a time limit set for the test in the assignment definition, a pop-up window warns students as they approach the time limit. They are not allowed to answer additional questions once they exceed the time limit; however, they can submit their work up to that point for grading.
- By default, the system does not establish a scheduled start or end time for assignments.

See Also:

Configuring Feedback Options (page 43)

Assignment Visibility

Clearing the **Visible** check box excludes the assignment from the list of assignments displayed to students on your class homepage.

Alternatively:

- In the **Assignment Editor**, you can access this option by clicking the **hide** option available when you hover your mouse over the assignment name.
- You can set the schedule for the assignment to exclude a particular period of time (which also restricts the assignment's availability to students).
- You can delete an assignment, but only if you are prepared to lose all of the results for the assignment from the **Gradebook**.

Setting Assignment Requirements: Limiting Student Access to Assignments

You can restrict access to an assignment so that only students meeting certain criteria can take the assignment. You can also create assignment requirements that refer to the current assignment, even if you are working with a new assignment that has not yet been saved. The assignment on which you are working is now listed along with all other available assignments in the **Criterion** specification box. This is useful when you want to ensure minimum competencies in prerequisite topics. For example, you can limit access to students who have already passed an earlier assignment.

To restrict assignment access:

1. From the **Assignment Editor** screen in the **Set Policies** tab, click **Advanced**. The **Set Policies - Advanced** screen is displayed.
2. To set criteria, click **Design**. A **Criterion 1** rule-based field opens. Specify the desired criteria.
3. **Save Changes to Criterion:** After you have specified the desired criteria, click **Back** to save changes made to the criteria and return to the **Set Policies** tab.

Criterion Options

- The first drop-down list gives a choice between has and has not.
- The drop-down list in the middle lists a range of states and actions, as shown.
- The drop-down list on the right lists all of the assignments for the class, including the assignment currently being edited.
- **Adding an OR Criterion:** Click the **Add alternative criterion** button immediately below the list of assignments. The program adds the list fields for another criterion.
- **Adding an AND Criterion:** Click the **Add additional criterion** button at the bottom right of the form, below the frame that encloses the criterion fields. The program adds another criterion group, in a separate frame.
- **Deleting a Criterion:** To delete an OR criterion, click on the **Remove criterion** button *inside* the frame for that criterion group. The program deletes the last criterion from the group. You may have to change the settings for the remaining criteria in the group to set the requirements that you want. To delete an AND criterion group, click on the **Remove criterion** button at the bottom of the form, below the last criterion group. The program

deletes the last group. You may have to change the settings for the remaining groups to set the requirements that you want.

Review & Finish Tab

Finishing and Saving an Assignment

Saving an assignment involves saving information in cached memory and in your class configuration.

To accept and publish the assignment you have created or modified, click **Finish** in the **Review & Finish** tab.

- If you have scheduled it for immediate availability, your new assignment is available for students on the **Class Homepage**.
- The system does *not* save partially completed assignments until you click **Finish** in the **Review & Finish** tab of the **Assignment Editor**. If you begin to create an assignment and then abandon it without clicking **Finish**, your additions and changes will be lost.
- If the server shuts down before you click the **Finish** button, your editing changes will be lost. When the server resumes, it will use the last saved version of your assignment data, and any unsaved changes you had made will be lost.
- To reduce the risk of this during long assignment editing sessions, you may want to periodically use the **Finish** button, then select your assignment from the main **Assignment Editor** list of assignments to start a new editing session. If you do this, it is recommended that you mark the assignment as *hidden* so that students cannot access it. Hide assignments from the main Assignment Editor page by hovering the mouse over the assignment name in the assignment list and clicking **hide**.
- After clicking **Finish**, you are returned to the **Assignment Editor** list of assignments for your class, where your new assignment is displayed in the list of all class assignments. Additions or changes you have made during this session are also displayed.

Canceling Assignment Creation

To cancel assignment creation:

1. In the **Review & Finish** tab of the **Assignment Editor**, click **Cancel**.
2. You are asked to confirm that you want to discard the assignment and modifications you have made.

To discard the assignment, click **OK**. You return to the **Assignment Editor** list of assignments. The assignment is not saved.

To continue editing the assignment, click **Cancel**.

Reviewing Assignment Details

In the **Review & Finish** tab of the **Assignment Editor**, you can review settings for your assignment and preview the assignment questions.

To change settings:

3. Click the appropriate **Assignment Editor** tab to return to the tab where the option is set. (If you click on the highlighted link for a policy, it will tell you where this setting can be accessed.)
4. Make the desired modifications and proceed to the **Review & Finish** tab.

4 Working with the Gradebook

4.1 Overview of the Gradebook

The **Gradebook** has four primary functions:

- Viewing and analyzing scores and statistics for students, assignments, and question items
- Reviewing student results
- Creating customized grade reports that reflect your grading policy. These grade reports can be saved or published for student viewing.
- Exporting grades to a file

In addition, from the Gradebook you can grade essay questions, add comments or adjust grades, add external assignments, and force grade active assignments.

Viewing Scores and Performing Gradebook Searches

The system automatically stores assignment session scores in the Gradebook. The Gradebook stores a variety of information for each student, such as the assignment start time, the time spent on the assignment, and the individual assignment question performance.

Use the Gradebook to view grade data organized by assignment, by student, or by question item. View the results online or export them to use in programs such as Microsoft® Excel. You can download the results of any Gradebook search or report to a file in comma-delimited format.

Search results are based on the data gathered by the system during student sessions, including statistical analysis of student and class performance. You start a search by selecting what information you want to include. The Gradebook makes it easy to get to the level of detail you need.

For example, you can begin by listing all student results on an assignment, and then click a student's name to display only that student's data. From the student record, you can then change a grade or view the student's answers for each question on an assignment.

Generating Grade Reports

You can define a grade report for your specified grading policy. For instance, you can group homework assignments and make them worth 30% of the overall grade, group quizzes (worth 30% of the grade), and make the final exam worth the remaining 40%. You could save this and publish it, allowing students to see their grade by your calculation.

Grade reports are also useful for providing snapshots of the grades to date or trying out alternative methods for weighing assignments. With options for designating extra credit and dropping the lowest grades, grade reports provide flexibility for you in assessing your class.

Grade reports allow you to:

- group assignments
- set criteria within an assignment group, such as "drop the lowest grade" and whether to ignore non-attempts or mark as 0
- designate weighting for each assignment in a group
- designate weighting for each assignment group in the overall grade
- designate an assignment group as extra credit
- save the grade report
- publish the grade report for student viewing

See Also:

Working with Grade Reports (page 56)

4.2 Performing Gradebook Searches

The **Gradebook** provides two types of searches of assignment data: class grades and item statistics.

- **Class Grades** — search for any number of assignments. For each assignment, the total grade for each student is displayed.
- **Item Statistics** — search for any number of assignments. For each question, item statistics are displayed (number of correct, partially correct, and incorrect responses as well as success rate, p-value, d-value, p-biserial, and r-biserial).

To perform a gradebook search:

1. From the Class Homepage, select **Gradebook** from the menu bar.
2. Select **Search** and the type of search: **Class Grades** or **Item Statistics**.
3. In the **Search Panel**, select the assignments you want to include in the search.
4. Specify additional search criteria.
5. In the **View Panel**, select the data to include in the search results.
6. Click **Search** to perform the search.

To start a new search, from the menu bar, select **Gradebook>Search** again, and then select **Class Grades** or **Item Statistics**.

Setting Search Criteria

The search panel is used to request data from the database. To perform a search, you must specify the assignment or assignments to include. You can select assignments by name or by assignment type. To include all assignments, select **All**.

If desired, customize your gradebook search using the options in the **Advanced Search Panel**:

- Classes — search within the current class or any child classes
- User — search for a user by login name
- List — specify whether to show all enrolled users or only users with grades
- Results — specify which user roles (students, instructors, and proctors) to include in results.
- Show Results — select from best, average, most recent, earliest, all (best), all (most recent), or all (earliest). If you let students attempt an assignment multiple times, you can select which result to include in your report. For example, if you select all (best), the search results display each attempt, but only the best grade is used when computing the mean, median, and item statistics.
- Progress — show completed assignments, assignments in progress, or assignments that need to be reviewed
- Date Range — specify a date range. Assignments completed within that date range are included.

Setting View Criteria

The view panel is used to specify what data is included in the results.

After performing a search, you can change the options in **View Panel** and refresh the view without performing another search.

Customize the display of the search results using these options:

- Rows — control the number of rows (of students) displayed per page
- Class Name — show the class name
- Style — indicate the presentation style for grades: number, percent, or letter grades
- Grade Data — display assignment details such as the date and time the assignment was started and finished, the time used to complete the assignment, and the number of attempts
- User Data — specify what student identification information to display
- Summary Data — display assignment summary data, such as total points, number of attempts, mean, and median for the assignment

You can export any generated table of grades to a file by clicking **Export to CSV**.

Note: The mean and median are calculated based on the search. For example, if the search is for the best score for each student, the median computation is computed from those grades only.

4.3 Searching for Class Grades

Search for class grades to show the results of specified assignments. For each assignment, the grade for each student is displayed. You can search for class grades for just the current class or including any child classes.

After performing a search, you can change the options in **View Panel** and refresh the view without performing another search.

You can export any generated table of grades to a file by clicking **Export to CSV**.

From the Class Grades, you can:

- View Assignment Grades by clicking on the name of an assignment
- View Student Grades by clicking on the name of a student
- View Student Assignment Details by clicking on any grade in the search results
- View Item Statistics
- Change the total points for an assignment

Viewing Assignment Grades

From the Class Grades, click the name of an assignment to access Assignment Grades.

This lets you view the question-by-question scores. The grade for each question (or question group) for each student is displayed.

Assignment Grades includes a summary of information on the assignment.

Table 4.1: Assignment Summary

Field	Description
Assignment Name	The name of the assignment. Click this link to collapse the details view.

Field	Description
Current Total Points	The total number of points.
Original Total Points	The total number of points originally assigned.
# Active Assignments	The number of assignments that are currently in progress.
Class Average	The average score of all assignments taken.
Last modified	The date and time the assignment was last modified.
# Attempts	The total number of assignments taken by students. This number might be greater than the number of students if the assignment allows for multiple submissions of work.
Average # Attempts	The average number of attempts per student.
Item Statistics	Click this link to view the item statistics for the assignment.
Histogram of Student Assignment Scores	Click this link to view a histogram of the assignment results. The histogram opens in a new window.

From the Assignment Grades, you can:

- Change the options in **View Panel** and refresh the view without performing another search.
- Export any generated table of grades to a file by clicking **Export to CSV**.
- Go to the Item Statistics for the assignment.
- View a histogram of student assignment scores.
- Change the total points on the assignment.
- View assignment results for a question group in the assignment.
- Preview a question on the assignment.
- Access question details for a question in the assignment.
- Access details for a question group in the assignment.

If the assignment includes question groups, these are indicated by a + symbol after the question number, such as Q1+. Click this link to see details on the items within this question group.

Viewing Question Responses

From the Assignments Grades, you can access Question Responses.

- In the Assignment Grades, click the name of a question, such as Q1, to view the student responses to this question.

In the Question Responses view, you can:

- View item statistics
- View student responses for each question. As part of this, you can view the instance of the question that the student received (for algorithmic questions).
- Add instructor comments
- Update student grades, if desired.

Viewing Student Grades

After performing a basic gradebook search, you can click the name of a student to access the student grades for that student.

The student record report first shows a summary for the student, including the number of completed, in progress, and passed assignments for the student, along with the number of assignments that need to be reviewed. For each assignment, the student score and total points are shown.

You can opt to show all results, or just the best, average, most recent, or earliest. For all of these except average, additional information for each assignment session includes session start and end times and duration.

You can click the **Details** link for any assignment to access the student assignment details.

Viewing Student Assignment Details

After performing a basic gradebook search, you can click any grade in the search results to view the detailed results for that assignment attempt.

The student assignment details report shows a summary of the results for this assignment attempt, including the student's score out of the total points, the session start and end times and duration. A summary for student is also shown, with the number of completed, in progress, and passed assignments for the student, along with the number of assignments that need to be reviewed.

In the Student Assignment Details view, you can:

- View the student's response for each question. As part of this, you can view the instance of the question that the student received (for algorithmic questions).
- Add instructor comments
- Update student grades, if desired.

Changing Grades and Adding Comments

From the Gradebook, you can modify student grades if needed. You can:

- Grade essay questions
- Change a student's grade on an individual question
- Leave comments for the student on an individual question

Changing Grades and Grading Essay Questions

You might want to change a student's grade for several reasons:

- The system does not automatically grade essay questions, so you must enter a grade for any essay questions included in your assignment. If an assignment contains questions that have not yet been graded, such as essay questions, then an icon ( ) appears to the left of the score.
- You might choose to raise a student's grade based on effort, improvement, or other subjective measure.
- You might choose to apply a curve, based on the performance of the entire class.

As you change grades, keep in mind the total and passing scores you have set for the assignment.

To change a student's grades:

1. From the Class Homepage, select **Gradebook** and click **Class Grades**.
2. Select the assignment with the grade(s) you want to change.
3. Click **Submit**.
4. From the resulting report, do one of the following:
   - Click the name of a student that requires grade changes. In the **Student Assignment Details** view, click the Details link beside the assignment that requires changes.
   - Click the name of a question that requires grade changes to view the **Question Responses**.
5. Each question is displayed, along with the student's response.

To update one grade:

- Locate the grade to be changed. In the corresponding **New Question Grade** text box, enter the new grade. The grade should be a number between 0.0 and 1.0, with 1.0 meaning fully correct.
- You can add a comment in the **Comment on Grade** text box. This region becomes writable after a new grade has been entered. This comment is for your reference, and is not seen by the student.
- Click **Update Grade** to save the change.

To update multiple grades at once:

- Select the questions using the check boxes on the left.
- Enter the new grade in the **Update selected grades as** field.
- Click **Update Grades**.

6. You can view a summary of the item's grade changes together with the accompanying comments by clicking **View History** beside the Comment on Grade text box.
7. You can add a comment for the student in the **Instructors Comment** text box, if desired.
8. Click **Update** at the top of the page.
9. You return to the student's record.
10. Click **Submit** to regenerate the statistical information.

- An asterisk appears beside the assignment name if you have edited the details of the assignment, for example, adding a comment or changing a grade.
- If the assignment is the original graded by the system, no asterisk appears beside the assignment name.

Adding Comments to Individual Graded Questions

You can add a comment for a student in the **Instructors Comment** text box. Students see these comments when the review their performance on the test. You can add personalized comments to help individual students understand their score and master the material.

To add comments to graded assignment questions:

1. From the **Class Homepage**, select **Gradebook** and click **Class Grades**.
2. Select the assignment for which you want to add a comment.
3. Click **Submit**.
4. From the resulting report, click the name of a student, or the name of a question.
5. Each question is displayed, along with the student's response.
6. Under each question is a **Instructors Comment** text box. Add a comment for the student in the desired box.
7. Click **Save Grades**.
8. You return to the student's record.

Students will see these comments when they review their performance on the test. You can add personalized comments to help individual students understand their scores and master the material.

Note: The comments you enter in this way are sent to individual students, and not provided as a part of question feedback. To enter question-level feedback (even response-specific feedback for multiple choice items), you must use the **Question Editor**.

See Also:

Configuring Feedback Options (page 43)

Changing the Total Points for an Assignment

To modify the total points for an assignment:

- In the search results in the Gradebook, click **Total points** in the left column.
- Enter the new total points for an assignment in the **Overridden** column.
- Click **Submit** to save the change.

4.4 Searching for Item Statistics

View item statistics for the questions in specified assignments. For each question, item statistics are displayed (number of correct, partially correct, and incorrect responses as well as success rate, p-value, d-value, p-biserial, and r-biserial).

Important: The system allows a variety of possible assignment applications for each question in the question repository, which means calculating traditional *item analysis* on questions may produce unexpected results. Remember that the analysis and statistics produced in the **Item Statistics** reports here are based on the limited use of question items in the specific context of the assignments you specify. If you specify assignments that are of more than one assignment mode or that do not have identical assignment settings, item statistics calculated here are unreliable. To produce reliable item analysis results, select only one assignment for **Item Statistics** or ensure that all selected assignments have identical rules and policies.

Table 4.2: Item Statistics

Success Rate	The **Success Rate** of an item is the average normalized score on that item. With each item graded between 0.0 and 1.0 (that is, normalized), the average score is computed over the set of all scores recorded for that item in the current assignment.
p-Value	The **p-Value** of an item measures the proportion of students who got the item correct. It is defined as the ratio of the number of fully correct responses to the total number of responses in the data set. That is, $$p = \frac{F}{n}$$ where n is the number of responses and F is the number of fully correct responses.
d-Value	The **d-Value** measures the discrimination of an item. To calculate it, the dataset is divided into two groups based on the scores on the entire assignment. The sets consist of those that scored greater than (or equal to) the median grade on the assignment and those that scored less than the median grade on the assignment. The p-Value for each set is calculated, and the d-Value is the difference of the p-Value for the high-scoring group and the p-Value for the low-scoring group. $$d = \frac{F1}{n1} - \frac{F2}{n2}$$ where n1 = number of assignments with scores $\geq$ median grade for the assignment n2 = number of assignments with scores $<$ median grade for the assignment F1 = number of fully correct responses on this item from the top-scoring set F2 = number of fully correct responses on this item from the bottom-scoring set
p-Biserial	The **p-Biserial** is a correlation coefficient relating performance on an item and performance on the total assignment. The p-Biserial correlation coefficient is an index of discrimination that measures the extent to which students who score high on the assignment tend to get the item correct and those who score low tend to get the item incorrect. $$r_{pb} = \frac{(Y_1 - Y_0)\sqrt{pq}}{\sigma_Y}$$

	where Y_0 = mean grade of assignments on which this item was answered incorrectly. Y_1 = mean grade of assignments on which this item was answered fully correctly. p = p-Value q = 1-p σ_Y = standard deviation for assignment grades
r-Biserial	The **r-Biserial** is a correlation coefficient relating performance on an item and performance on the total assignment. The r-Biserial correlation coefficient is an index of discrimination that measures the extent to which students who score high on the assignment tend to get the item correct and those who score low tend to get the item incorrect. $r_b = \frac{\left(Y_1 - Y_0\right)\left(\frac{pq}{Y}\right)}{\sigma_Y}$ where Y_0 = mean grade of assignments on which this item was answered incorrectly. Y_1 = mean grade of assignments on which this item was answered fully correctly. p = p-Value q = 1-p σ_Y = standard deviation for assignment grades Y = height of the standardized normal distribution at the point z, where P(z'<z)=q and P(z'>z)=p **Note:** Since the r-Biserial is scaled by a parameter that is based on a normal distribution, it can result in values outside of -1 to 1 if the scores are not normally distributed.

4.5 Working with Grade Reports

You can create grade reports that reflect your class grading policy, provide a snapshot of grades-to-date, test out alternative methods for weighing assignments, and so on.

A grade report is defined by selecting assignments, putting them into report groups, and defining the weighting for each group. You can control the grading policies that determine how grades.

Creating a New Grade Report

To create a grade report:

1. From the Class Homepage, select **Gradebook**, then select **Grade Reports**.
2. Click **New** to define a new report.
3. Specify a name for the grade report.
4. Then, click **Add Group**.
    a. Specify a name for the report group.
    b. Select the assignments for the group.
    c. Specify the weight of this group in the overall grade computation.
    d. If desired, adjust additional settings.
5. When finished, click **Save** to save the group settings.
6. Add more groups as needed.
    - Within a grade report, an assignment can only be in one group. Thus, any assignments that were already placed in a group will not appear in the **Available Assignments** list for other groups.
    - The weightings of the groups in a grade report (not including any Bonus assignments) should add to 100%.
7. You can change the order of the groups listed in a grade report. To change the order, select a new position number from the drop-down menu of the assignment you want to reposition. Once selected, all affected assignments will adjust their position accordingly.
8. To publish a grade report so students can view the report, select the **Visible** check box.
9. Click **Save** to save the grade report.

For a report group, you can set the following settings:

- Group Name — the name of the group in the grade report
- Weight — set a weighting for the group in the overall grade for the report
- Drop Lowest — drop the lowest *n* grades in the group
- Weight Assignments — weight the assignments in the group equally (normalized) or by using the total points for each assignment
- Non-Attempts — mark non-attempts as 0 or ignore
- Use — if an assignment is attempted multiple times, specify whether to use the best, most recent, earliest, or average grade
- Bonus — mark a group as extra credit. If a group is marked as Bonus, its weight is not used to reach the cumulative weight of 100% for the grade report.

Viewing a Grade Report

To view a grade report:

10. From the Class Homepage, select **Gradebook**, then select **Grade Reports**.
11. Click **Load** to view the report.
12. In the View panel, select data to include in the results and then refresh the view of the grade report.

In a grade report, bonus assignments are indicated with the bonus icon (⭐).

Publishing a Grade Report for Students

You can publish a grade report for student viewing. To publish a grade report, edit the grade report and select the **Visible** check box.

4.6 Exporting Reports and Performance Data

You can export data from any gradebook search or for any grade report you generate in the **Gradebook**. Exported performance data files are in comma-delimited (CSV) format and can be opened in a variety of applications (for example, Microsoft Excel) and are easily imported into many third-party gradebook programs.

To export data from the Gradebook:

1. In the gradebook, perform a search or load a grade report.
2. Click **Export to CSV** to open or save as a comma delimited file.

4.7 Adding an External Assignment

You can add external assignment grades to your Maple T.A. gradebook. There are two ways to add external grades, by uploading a roster of grades, or through the gradebook interface.

Adding External Grades using a Roster

Grade rosters are an efficient way to add a large number of grades at once. By uploading a file, you can add multiple external assignments at once.

To upload external grades:

1. Create a text file in the specified format.
   - The first column identifies the users. This can be login names, email addresses, or student IDs.
   - The remaining columns provide assignment data. Assignment data consists of the following, in order: the assignment name, the total points, the passing score, and the student scores.

   Save the file as a text file, with fields delimited by comma or tab characters.
2. From the Class Homepage, select the **Gradebook** menu and then select **External Import**.
3. Click **Browse** to find the text file containing the grade roster.
4. Specify whether the file is comma-separated or tab-separated.
5. Click **Submit**.
6. Preview the imported data. Click **Save External** to save the external assignments to the gradebook.

You can now view this data in the Gradebook.

Adding External Grades Individually

To add an external assignment manually:

1. From the Class Homepage, select the **Gradebook** menu and then select **External**.
2. Enter the assignment name, total points, and a passing score.
3. For each student, enter the score they received on the external assignment, and add an optional comment.
4. Click **Save** to save this data to the gradebook.
5. Click **Save & Add Another** to add another external assignment to the gradebook.

You can now view this data in the Gradebook.

Modifying External Assignment Grades

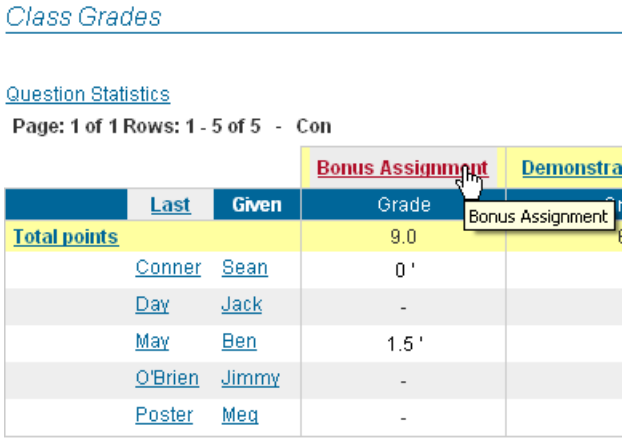
You can modify an external assignment in the gradebook.

1. In the gradebook, perform a search for the assignment.
2. Click the name of the assignment to edit the external grades.
3. Modify the assignment name, total points, passing score, if desired. Modify student grades or comments.
4. Save the results.

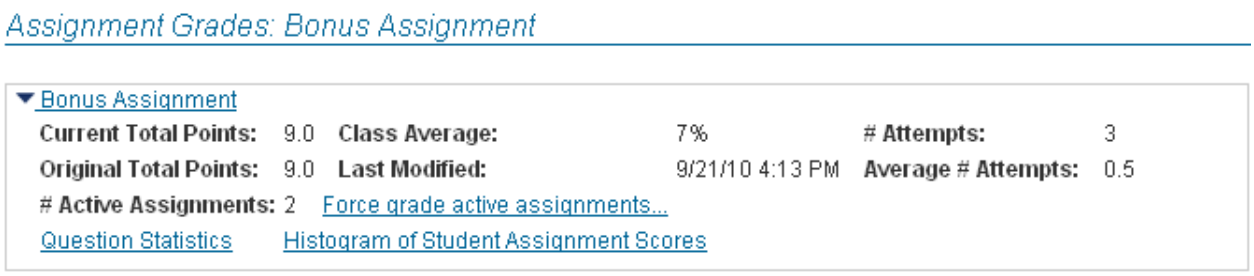
4.8 Forced Grading

To force the grading of an assignment:

1. From the Class Homepage, select the **Gradebook** menu and then select **Class Grades**.
2. Select the **Assignment** name and, in the **Progress** list select **In Progress**.
3. Click **Submit** to perform the search. An in progress assignment is indicated by the ' symbol beside the grade.
4. In the search results, click the assignment name link.



5. In the Details area, click the **Force grade active assignments ...** link.



6. The users that are currently taking the assignment are displayed. Click **Force Grade**.

Important: If a student is actively in the process of taking an assignment, you cannot force grade the student's assignment. If you attempt to force grade an assignment that any active users are currently taking, a list of active users is displayed with the message that those assignments will not be force graded.

See Also:

Changing Grades and Adding Comments (page 53)

5 Course Modules

Course Modules allow you to share your class content with others, including related assignments and any web resources (for example, images) used by your questions and assignments.

5.1 Course Modules Overview

When you create a course module (CM) the system packages the class elements you have identified, producing a single ZIP file that can be archived or redistributed to other instructors using the system. The instructors can have classes on the same server or on other Maple T.A. servers.

It is an ideal method for content management of entire classes full of related questions, assignments, and web references, and allows you to include all or only specific parts of those classes in the CM package.

Using CMs is similar to the share class capability in that it allows you to share your class content; however, once a CM is distributed and installed in a new class location, it does not maintain any synchronization with the original source class or its materials.

CMs ensure transportability of content from one class to another, archiving of entire class contents, and even portability of entire classes from one Maple T.A. server to another.

When you create a CM, the web reference URLs inside your questions are converted to a portable format, so that when you install the CM in a new class, these can be automatically set to the new class web site address by the system as part of the installation process.

Access **Course Modules** from the Class Homepage by selecting the **Content Manager** menu and then selecting **Course Modules**.

In the **Course Modules** screen, a list of the currently installed CMs for your class is displayed along with these options. (The list is empty initially.) See **Figure 5.1**.

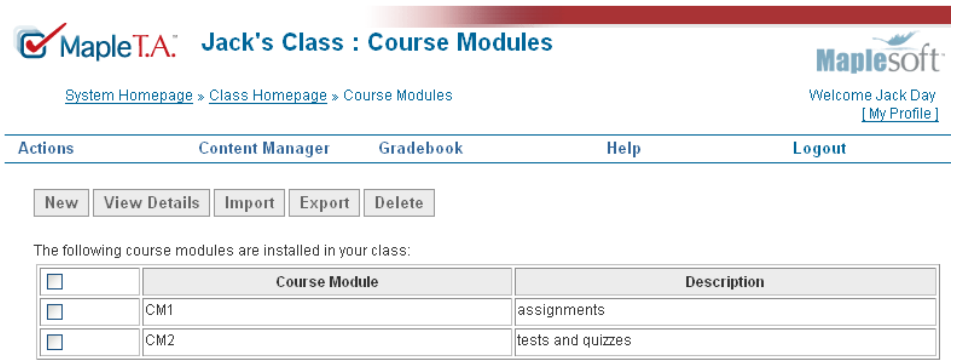


Figure 5.1: Create, Import, Export, or Delete Course Modules

New - Create a new CM

View Details - Select one or more existing CMs from the list and view their individual contents: questions, assignments, and resources

Import - Upload a CM from your hard drive

Export - Select one or more existing CMs from the list and save them to your hard drive

Delete - Select one or more existing CMs from the list and delete them

The advantage of using course modules rather than question banks or featured classes to manage content between classes on one or more Maple T.A. servers is that the questions, assignments, and web resource folders are entirely portable and can be installed in a new class (even at another institution) very easily. After you create and export a course module, you can send it to other instructors who can upload it into their class and use it as-is or alter it to suit their needs.

Technical Information: Expected Behaviors of Course Modules

1. CMs encapsulate from the source class and allow you to reinstall questions in the destination class.
    - Relative paths to resources are preserved and reinstalled.
    - All question types (including those requiring outside resources) are preserved. **Note:** Different releases of the system may support different question types and earlier system releases may not support all question types.
2. CMs encapsulate from the source class and allow you to reinstall assignments in the destination class.
    - Assignment behaviors and policies are preserved.
    - Links to all source content and related resources are preserved.
3. CMs encapsulate from the source class and allow you to reinstall resource objects in the destination class.
    - Relative paths are preserved for folders and subfolders of the class web site .
    - Underlying folder structure is preserved.
    - Connections with questions and assignments (through annotations) are preserved.

See Also

Shared Classes Overview (page 13)

5.2 Creating a Course Module

To share content using a course module, first create a course module by choosing the items you wish to include. After you have created the course module, save the course module to your computer as a ZIP file. You can then share the course module with others.

To create a new CM:

1. From your Class Homepage, select the **Content Manager** menu and then select **Course Modules**.
2. Click the **New** button.
3. A table displays all the questions, assignments, and web folders holding resources (for example, image files) for your class. Select the course components to be included in your new course module.
   - You can highlight multiple items in a list using **Shift**-click (for contiguous selections) or **Ctrl**-click (for multiple individual selections).
   - Use the button to add your selections to the list of included elements for your new course module.
   - You can include any combination of questions, assignments, and web folders in your selected list.
   - To select web resources (for example, images and other files that are referenced by your questions and assignments), select the folders that contain your resources. Subfolders are automatically selected.
4. When you are satisfied that the list on the right contains all the elements to be included in your new course module, click **OK**.
5. Specify a name for your course module. Optionally, you can enter descriptions for each course component. You can also select descriptive names for any web folders you have selected. Question and assignment names cannot be changed.
6. Click **OK** to create your Course Module according to the specifications you have indicated. You return to the **Course Modules** main screen and your new Course Module has been added to the bottom of the list of installed course modules for your class.

Note: Click **Cancel** at any time to return to the **Course Modules** main page.

See Also:

Exporting a Course Module (page 64)

Adding and Deleting Elements within a Course Module (page 66)

Viewing Details of a Course Module (page 66)

Course Module Creation and Shared Class Inheritance of Questions and Assignments

CMs can only include content that is local to the source class.

- Content inherited from another class cannot be included.
- The course module creation screen will not show any inherited content.

See Also:

Shared Class Content Updates (page 13)

5.3 Exporting a Course Module

Click **Export** to save the selected CM from your source class to your hard drive. This is a required step if you want to redistribute or install your CM into a new class.

1. Create a CM.
2. Select the CM you want to save to your hard drive by selecting the check box adjacent to it in the list of available CMs in your class.
3. Click the **Export** button. The next page displays all the related URL information that is referenced in the CM you have selected. You will see a message about the URLs as follows.

   > Your questions will be scanned for URLs beginning with the following strings. These references will be replaced with a system-independent string that will make your module relocatable to classes on other servers.

   When the CM is saved, it replaces the source class-specific URL information with relative references that are then localized upon reinstallation. Typically you will not want to make changes to the system default behavior for URL replacements.
4. Click **OK** to continue. The course module is prepared for download.
5. When prompted, start the download. A file download box will appear, and the CM is saved to the default name **Cm.zip**.
   Note: It is strongly recommended that you rename the CM, replacing the default name **Cm.zip** with a more meaningful name.

When the file download/save operation is complete you are returned to the main **Course Modules** screen.

5.4 Importing and Installing a Course Module

Note: In the following discussion, the term *destination class* refers to the class that you want to install the course module into and the term *source class* refers to the originating class for the CM contents.

To install a new course module into the destination class:

1. Ensure that you have access to the saved course module from the source class on your hard drive, a network drive, or an Internet resource.
2. Click the **Import** button in the **Course Modules** screen.
3. To select a course module, click **Browse**.
4. In the **Choose File** dialog, select the course module on your hard drive, a network drive, or an Internet resource, and click Open.
5. Click **Import** to initiate the file upload. (The speed of the upload process depends on your internet connection.)
   - Elements contained in your uploaded course that are also contained in your class are highlighted in yellow background with red type, with a warning message:

     > **WARNING:** *The course module you have uploaded contains resources which will overwrite components that you already have installed. The components which will overwrite your existing course components are highlighted in the table below. If you do not want these course components erased and replaced with new ones, press Cancel, or ensure that they are not selected in the table below. To proceed, select the components that you want to install from the table below and press Install.*
6. At this point you can install the new components in your class, overwriting the original components, or you can clear the check box adjacent to the duplicate element in the CM details prior to proceeding. You can also select or

clear other elements of the course module for installation into your class. The default installation behavior is to install all course module elements; however, by clearing individual parts you can choose the specific elements you want to install. Click **Install** to copy and install all the selected course module elements into your destination class.

7. The new course module appears in the list of available course modules for your class, and all of the elements you selected appear in their respective areas throughout your class web site. Questions are listed in the **Question Repository**, assignments appear in the **Assignment Editor**, and web resources are displayed in the **Web Site Editor**.

Requirements

- Course modules are files with a .zip extension.
- If you attempt to upload a ZIP file that is *not* a CM, the system generates the following error message and you must make another attempt:

  This file does not appear to be course module (it has no manifest file).

- Each server has a maximum file size setting for uploads that is established by your system administrator. (The system default setting is to allow uploads of up to 1 MB.) If your file upload repeatedly times out, or if you receive a message informing you that your upload exceeds the server limit, contact your system administrator.

Additional Notes

The most common element that will be flagged as a duplicate is the base class web site folder, **web_folders**. This is a common default destination for many web resources; however, because it is a folder name established for every course, you may already have saved your own resources to the class web site folder in the destination course. If this is the case, you can either:

- Create a new subfolder in your destination class web site and reinstall your resources in that subfolder
- If you have access to the source class for the CM, you can return to the CM's source class and move the resources in the base folder of that class web site to a subfolder, and then recreate the CM so that it will not overlap with your destination class folders upon upload.

Course Module Installation and Shared Class Inheritance of Questions and Assignments

If you overwrite or edit or install a CM that has a new version of a question with the same name as the old, **you permanently break the link** to the *parent* class version of the question. If, in the future, you want to refresh the content of your class to match that in the parent class or restore the link to the parent class version of the content, you must save your new, edited question under a different name and then delete the version that has the same (original) name as the original in the parent class question, or delete the edited/modified version of the question. The parent class restores the matching (original) question automatically.

Note: If you modify questions or assignments that you inherit from a shared parent class, there is no risk that the parent class version will overwrite your modified version, unless you specifically delete the modified question or assignment from your child class. This means that if you make changes to the inherited questions or assignments and save them to the original name, your changed versions sever the relationship with the parent asset and you will not receive any future updates from the parent class automatically. Therefore, you must stay in communication with the instructor in your shared parent class if you want to be notified of changes to parent class questions and assignments.

See Also:

Shared Class Content Updates (page 13)

5.5 Viewing Details of a Course Module

View Details allows you to inspect the individual elements contained within a selected CM. You also use **View Details** as the first step to deleting parts from within a CM.

1. Select the check box adjacent to the CM you want to work with.
2. Click the **View Details** button on the **Course Modules** screen.
    - A page is displayed that shows the questions, assignments, and web resource folders that are contained in the CM.
    - A description of each component may also be included (depending on whether the creator of the CM included this information).
    - If you have selected multiple CMs for viewing, each is displayed in sequence, with the above information posted.

5.6 Adding and Deleting Elements within a Course Module

Adding Course Module Contents

The current release of the system does not allow you to add CM contents to an existing course module. Instead you should create a new course module that contains all the elements you want.

Deleting Course Module Contents

You may want to remove specific questions, assignments, or web folders from within a CM.

1. From the **Course Modules** screen, select the CM you want to work with from the list of available CMs in your class. You can select multiple CMs for this action.
2. Click **View Details** to proceed to a page that lists all the questions, assignments, and web resource folders that are contained in the selected CMs.
3. Select the components to be deleted, and click **Delete**.
4. You can select all the questions, assignments, or web folders included in the CM by selecting the check box adjacent to the **Name** column headings.

Important: When you use this method to delete elements from a CM, the **system deletes the selected element**: questions, assignments, or web resource folders are physically deleted from the system and are unrecoverable.

Note: If the elements you have deleted from one CM are used in another CM installed in your class, they will *not* be permanently removed from your course, but will be removed from only the selected CM.

To delete only the index file containing the record of the CM contents from the list of the available CMs, click the **Delete** button.

Why Delete Only Part of a Course Module

If you install a CM given to you by a colleague, but want to duplicate only certain parts of the contents of that CM, for example, install only questions and web folders but not install the related assignments, you first install the entire CM, and then use the steps outlined above to remove the assignments.

5.7 Deleting a Course Module

The **Delete** button removes the *index* file that stores the record of what is included in the course module without actually deleting the class elements. Deleting the course module index file removes the record of the CM from the list of available CMs but does not actually remove its contents from the system.

To delete the course module *index* from the list of available course modules in your class:

1. Select the CM you want to remove by selecting the check box adjacent to it. You can select multiple CMs for this action.
2. A warning message is displayed. To remove this course module index from the list of installed course modules in this class, click **OK**. The CM is removed from the list of available CMs for this class.

To delete the *contents* of a course module, you must delete individual components, as described under **Adding and Deleting Elements within a Course Module**.

5.8 Relative Web References in Course Modules

When your questions are saved, Maple T.A. does a global *search and replace* for the strings listed.

```
http://server.name/Web/yourclass/Public_Html/
http://server.name/classes/yourclass/
../classes/yourclass/
```

It replaces all of them with:

```
__BASE_CLASS_URI__
```

When the questions are uploaded, the string is translated to:

```
http://server.name/classes/yourclass/
```

so that your image references are correct.

If you add more strings to the list, they must be replaced with:

```
__BASE_CLASS_URI__.
```

The strings listed are the most common ones for people to use in addressing images that are stored in their class, but it is possible that you have used a different addressing scheme.

You might also want to delete strings if there are addresses you do not want to be replaced. For example, if you intend to always refer to the copies of images stored on that particular class (that is, you are not copying the images), you might want to leave the URLs like **http://server.name/classes/yourclass/** in place.

6 Administering Maple T.A.

6.1 Changing Your Password

1. From the System Homepage, click the **My Profile** link located above the **Logout** link.
2. Click the **Password Update** button.
3. Enter your new password, confirm your new password, and click **Submit**.

6.2 Editing Your Profile

You can change your password, but only an administrator can change your profile information.

6.3 Deleting Student Records

To delete a student record:

1. From the Class Homepage, select the **Actions** menu and then select **User Manager**.
2. From the **Actions** menu, select **Remove Users**.
3. Select the check box next to the user(s) you want to remove from the class. Click **Remove**.

Important: Removing a user from your class removes all of that student's records from the Gradebook.

See Also:

Class Rosters (page 11)

6.4 Security Issues to Consider

It is important to protect your **Gradebook** and test questions. Review the following security suggestions for Instructor Passwords, Student Account Management, and Proctored Exams.

- *Instructor Password Security (page 69)*
- *Student Account Tricks and How to Foil Them (page 70)*
- *Secure Proctored Exams (page 70)*

Instructor Password

- Change your password often.
- Never allow a student to see it. The instructor password grants access to all the grades in your class, and enables you to change grades. Use caution when entering it in front of students.
- Your password is stored on the server in plain text. This means that the system administrator can read it. Therefore, do not use a password that cannot be disclosed to a third party.
- When you log in, the time of your last login is given under your user profile. Check this every time to determine if there are any unaccounted logins. Access your profile by clicking the **My Profile** link above the **Logout** link.

Student Account Management

1. Some students will log in to an assignment once or twice and click **Grade** without attempting any questions, to make a printout of the feedback. Using this feedback, they complete the assignment. There are three main solutions:
   - Restrict the number of retakes allowed. For details, see *Setting the Maximum Number of Attempts (page 45)*.
   - Set the feedback level to score only rather than giving out answers or solutions. (This approach penalizes the good students, too.) For details, see *Configuring Feedback Options (page 43)*.
   - If you are writing and using your own questions, you can edit the Comments fields and mask the correct answer. For details, see *Configuring Feedback Options (page 43)*.
2. Consider locking your class after an initial sign-up period to prevent registration of students not in the class and creation of fake accounts. You can also delete unwanted accounts. For details, *Locking and Unlocking Class Registration (page 10)* and *Deleting Student Records (page 69)*.

Proctored Exams

1. Proctors can authorize students to take and grade assignments. This usually depends on class size and the number of teaching assistants available to act as proctors. However, this double-checking approach keeps the environment totally controlled.
2. Proctors can use the **Proctor Tools** utility to avoid entering their password in front of students.

For more information, see *Proctored Exams (page 35)*.

7 Integrating Maple T.A. with an LDAP Server

7.1 LDAP Overview for Instructors

The LDAP integration in Maple T.A. allows you to integrate Maple T.A. with your campus database for user authentication, making the instructor's class administration even easier. LDAP integration allows Maple T.A. users to log in with a username and password that are managed by your university's IT group in a central repository -- an "LDAP Server." Your Maple T.A. system administrator can give you more information on whether LDAP is used at your institution.

What LDAP means to the Maple T.A. course administrator:

1. There is no student "Maple T.A. username" or "Maple T.A. password." Students only need their university username and university password to log in to Maple T.A.
2. As the Maple T.A. administrator, you don't have to worry about usernames, passwords, or resetting passwords. Your IT group will do that.

Basic user profile information comes from the LDAP directory. Your system administrator configures which information in the LDAP profile will be made available to Maple T.A.

7.2 Login and Forgotten Passwords with LDAP

Login

Anyone in the LDAP directory can use Maple T.A. The user simply logs in to Maple T.A. using his university username and password.

1. A user enters his or her username and password.
2. Maple T.A. presents these credentials to the LDAP server, and the LDAP server performs authentication.
3. If the credentials are rejected by the LDAP server, login is denied. If the credentials are accepted,
    a. If the user is already known to Maple T.A., profile information is acquired from the LDAP directory. Maple T.A.'s database is updated to reflect any changes.
    b. If the user is not known to Maple T.A., the user is added to the Maple T.A. database, retrieving profile information from the LDAP directory. Maple T.A. then displays a validation page for the user. The user is given the opportunity to add profile information not supplied by the LDAP directory, and must confirm the final profile information.

The login username and password are case-sensitive. You may use any combination of upper and lower case. Maple T.A. stores the username as found in the LDAP directory and that username will appear in web pages and reports.

Forgotten Passwords

If a user forgets his or her password, the user must contact your university's IT group. In an LDAP environment, Maple T.A. does not manage passwords, and the Maple T.A. course administrator cannot reset or retrieve passwords.

7.3 User Manager - Add Users with LDAP

The Add User action on the Administer Users page allows you to locate individuals in the LDAP directory who are not in Maple T.A. and add them to Maple T.A. The effect is the same as if the user had logged in.

You can search for users by attribute: username, first name, last name, or email address. You can use a * as a wildcard. For example,

```
Last Name an*
```

will display all names that start with "an" that are in the LDAP directory but not in Maple T.A. These searches are not case-sensitive, so **an*** produces the same results as **An***.

Once you have created a list of users in this way, you can select those you want to add to Maple T.A.

7.4 User Manager - Import with LDAP

When you import users in an LDAP Maple T.A. environment, user information is verified and updated from the LDAP directory. The roster import file must follow the specifications for Roster File Upload. The usernames in the import file are used to match users to the LDAP directory.

When you import users with LDAP:

1. The LDAP server is consulted to verify each user is in the LDAP directory. Users that are not in the directory will not be imported.
2. Matching of users is done by username.
3. Profile information from the LDAP directory will override any corresponding information in the import file.
4. Require User Validation, if included, is ignored with LDAP. No validation is necessary on the user's next login.

8 Authoring Methods for Questions

8.1 Advanced Instructor Help

Question Repository and Question Types	• Question Types • Math and Science Question Types • Maple™ Question Types • Question Repository
Question Authoring and Content Production	• Authoring Methods • Using the Question Editor • Avoiding System Time-outs during Authoring Sessions • Using Plain text Script Files • Using LaTeX

8.2 Navigating the Question Repository

The Question Repository is used to create, organize, and search for questions.

To open the Question Repository:

- From the **Content Manager** menu, select **Question Repository**.

The **Questions** menu provides the following options.

- Select **New Question** to start a new question.
- Select **Import Question Bank** to upload a question file from your hard drive.
- Select **New Search** to search the Question Repository for questions.

For more information on the question repository, see *Question Repository (page 114)*.

8.3 Overview: Authoring Methods for Question Banks

There are three methods for authoring question banks in the system.

- Question Editor — The Question Editor, also called QE, is the primary question authoring and editing method for small to medium projects involving a limited number of questions. See *Authoring: When to Use the Question Editor (page 75)*.
- Maple T.A. .**qu** plain text script files — These are plain text script files containing questions formatted according to the Maple T.A. syntax. These can contain HTML, MathML, and other standard web content. See *Authoring: When to Use Maple T.A. Plain Text Script Files (page 75)*.
- LaTeX files that can produce math-intensive questions formatted to Maple T.A. specifications — Requires you to have a text editor and LaTeX. Registered users can download macros. See *Authoring: When to Use LaTeX (page 75)*.

For project planning information, see *Planning Your Question Bank Project (page 75)*.

Question Banks

Independent of the authoring method you use, a Maple T.A. question bank file is produced. Question bank files are plain text files, with a **.qu** extension, structured according to the Maple T.A. question bank syntax.

Uploading files

If you use the **Question Editor**, the files are automatically saved with a **.qu** file extension. However, you can upload a plain text file with any file extension, for example, **.htm**, to the Maple T.A. system.

File Formats: Browsers and Platforms

With Maple T.A., you are essentially creating web pages. Not all file formats are supported by all browsers or will run on all platforms. If adding an image file, make sure that it is widely supported by various browsers such as **GIF**, **JPEG**, or **PNG**. Do not, for example, use bitmap image files in Maple T.A.

Which Method is Best for my Project?

Choose the best method for your project.

• To edit a few existing questions	Use the online Question Editor
• To reorganize question repository topics and questions within topics	Group questions in the question repository
• To work with algorithmically generated variable data in my questions	Use the Algorithm Designer within the Question Editor
• To work with math-intensive content	See Authoring Mathematical Questions
• To edit or author many questions	If you are an experienced LaTeX user, you can use LaTeX to quickly generate questions.
• To work with information fields	Use the Question Editor
• To work with symbolic expressions (chemistry, foreign languages, mathematics, logic)	Use LaTeX or the Question Editor
• To create specific advanced question types	Use LaTeX or plain text script files
• To use Maple to grade or generate plots in questions	See Maple-graded Question type and Questions with Plots

See Also:

Authoring: When to Use Maple T.A. Plain Text Script Files (page 75)

Authoring: When to Use LaTeX (page 75)

Authoring: When to Use the Question Editor (page 75)

Planning Your Question Bank Project (page 75)

The Algorithm Designer (page 80)

Authoring Mathematical Questions (page 179)

Maple-graded Question Type - Overview (page 140)

Questions with Plots (page 148)

8.4 Project Elements

Authoring: When to Use the Question Editor

Use the **Question Editor** to:

- Create new questions or edit existing questions. The **Question Editor** allows you to author many varieties of questions in the system.
- Author a limited number of new questions. (Scripting and LaTeX may be faster for large projects.)

Use another method when authoring advanced question types involving interaction with, for example, applets.

See Also:

Online Authoring in the Question Editor (page 92)

Authoring: When to Use LaTeX

Use Question Banks Created from LaTeX Files

- For an alternative to plain text script files when authoring several advanced question modes (for example, dynamically rendered plots in questions, algorithmically generated labels in figures, matrices, and dynamic graph sketching)
- You must have LaTeX expertise and technical knowledge. You must use the macro package and web-based conversion service (http://www.maplesoft.com/products/mapleta/latex2ta/).

See Also:

LaTeX Authoring Overview (page 247)

Authoring: When to Use Maple T.A. Plain Text Script Files

Use Maple T.A. .qu Plain Text Script Files

- For full access to the power and flexibility of the system question banks and algorithm syntax contained.
- For authoring all question types
- For full support of HTML, including image references, tables, and JavaScript

To write script files, use a text or HTML editor.

See Also:

Using Script Files to Author Questions (page 283)

Planning Your Question Bank Project

To start a new project:

Create a question bank. Because of the range of question types supported by Maple T.A., you can include a much wider range of classroom-tested assignment materials, including sophisticated free response questions for Mathematics and Science.

Once you have selected the source for your questions, adapt the source question material into Maple T.A. questions.

- Decide what type of Maple T.A. question mode works best for your source material.
- Decide what feedback, comments, hints, solutions, and related links are appropriate.
- You may also consider the range of tolerance, margin of error, and required format for correct answers.

Use information fields to tag your questions with information indicating authorship, level of difficulty, key topic, related topics, page reference, or other outside link information. Because the information fields are entirely user-specified, consider how you might want to use your questions, interrelate them in assignments, and what performance reporting dimensions might be useful.

If your question takes advantage of the system's powerful algorithmic variable generation capabilities, you must use appropriate variable definitions, data ranges, parameters, and other limiting conditions for each variable.

- Maple T.A. is automatically configured to recognize equivalent responses for numeric, algebraic, and numeric with unit dimension student responses; however, you can provide a question-bank specific table of equivalent units or expressions as well.

See Also:

Question Types Available in the System (page 131)

Setting a Margin of Error in Non-numeric Questions (page 184)

Tutorial: Displaying Answers vs. Formulas for Answers (page 212)

Controlling Display of Numbers within Questions (page 239)

Overview of Algorithmic Question Capabilities (page 217)

Default Table of Physical Unit Equivalents (page 205)

8.5 The Question Editor

Creating a New Question or Editing an Existing Question

The **Question Editor** allows you to:

- Create questions
- Edit existing questions
- Add algorithms, hints, comments, feedback, solutions, and information fields to questions

To create a question:

1. In the **Question Repository** page, select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. Enter the content of your question.
   (See *Question Authoring in the Question Editor (page 92)*.)
3. Once you have completed the question, click **Finish**.
4. Choose the groups (if any) that you wish the question to belong to and click **Submit**.
5. A preview of the question is displayed. Provide an answer and click **Grade** to see how the question is graded. Click edit to continue refining the question if necessary.

To edit a question:

1. In the Question Repository, search for the question.
2. In the right pane, click the **Edit** icon ( ) beside the question.
3. Click the **Edit** button.
4. Make your changes to the question.
5. Click **Finish** to save your changes.

Cloning Questions

To add a question that is similar to an existing question in the repository, you can save time by cloning the existing question and then editing the clone.

To clone a question:

1. In the Question Repository, find the question you want to clone.
2. Click the clone icon () beside the question. The question has now been cloned and the clone can now be edited.

See Also:

Sharing Questions (page 117)

Avoiding System Time-outs during Authoring Sessions

While you are working on a question, the system *client* holds your changes to the working copy of the active question. While authoring or editing a question, if you click **Finish**, the system places a copy of your question in the class web site cache.

If you do not interact with the system for about 30 minutes, the system logs you out of your session and purges the question data (losing any changes that have not been added to the web site cache). When you log in, you revert to the *cached* version of your question.

If the system *server* is restarted, the cache is purged. You revert to the last installed version of your question. You can also upload a saved version of your question from your hard drive to your client.

Important: If you stop interacting with the system for more than a few minutes, it is recommended that you save your question to the repository

The **question persistence** time is controlled by the system administrator. To change its value, contact your administrator.

Adding and Editing Comments

When editing a question in the **Question Editor**, you can add or edit the comment.

To add or edit a comment:

1. On the **Question Editor** screen, click **Add** or **Edit** in the **Feedback** area.
2. Enter or edit the comment in the Text Editor. Comments can include HTML-formatted text, JavaScript™, symbolic math expressions, algorithmic variables, inserted objects, and more. To add or edit HTML-formatted text, click the **Source** button (Source) in the menu bar.
3. Click **Save**.
4. To add your changes to the cached question, proceed to the **Question Display** screen by clicking **Next**, and then click **Finish**.

See Also:

Text Formatting in Questions (page 107)

Comments and Performance Feedback (page 129)

Using the Comment Field in a Maple Question (page 294)

Adding and Editing Algorithms

For every question type, you can use algorithmically generated variables in question statements, answers, hints, or feedback. Using variables in questions you can generate variations on a single *template* question.

To create a question that uses randomly generated data in a question, you must define variables.

Maple-generated plots to be included in questions are also defined in the Algorithm Editor using the plotmaple command. When you define a plot in the Algorithm Editor, you can use **Preview** to see the plot image, verify the plot statement is correct, and edit it if necessary. For more information, see *Plotting Using Maple (page 201)*.

When editing a question in the **Question Editor**, you can add or edit algorithms.

To add or edit algorithms:

1. In the **Question Editor** screen, click **Add** or **Edit** in the **Algorithm** area.
2. The Algorithm Editor opens. See **Figure 8.1**. The buttons above the algorithm text box perform the following tasks.
   - **Cancel** discards your additions or edits to the algorithms and returns you to the **Question Editor** screen.
   - **Clear** removes *all* variable statements in the algorithm region.
   - **Show Designer** displays the Algorithm Designer. After you click the **Show Designer** button, the system replaces it with the **Hide Designer** button. Clicking the **Hide Designer** button hides the **Algorithm Designer**. Only one of the buttons is visible at a time.
   - **Refresh** evaluates your variable definition and control statements, and returns a value for each variable in the **Variable/Value** region located below the algorithm text region. If you define a plot in the Algorithm Editor, the plot image is displayed. You can verify the plot statement is correct, and edit it if necessary.
   - **Save** saves your additions or changes to the algorithms and returns you to the **Question Name & Type** screen.

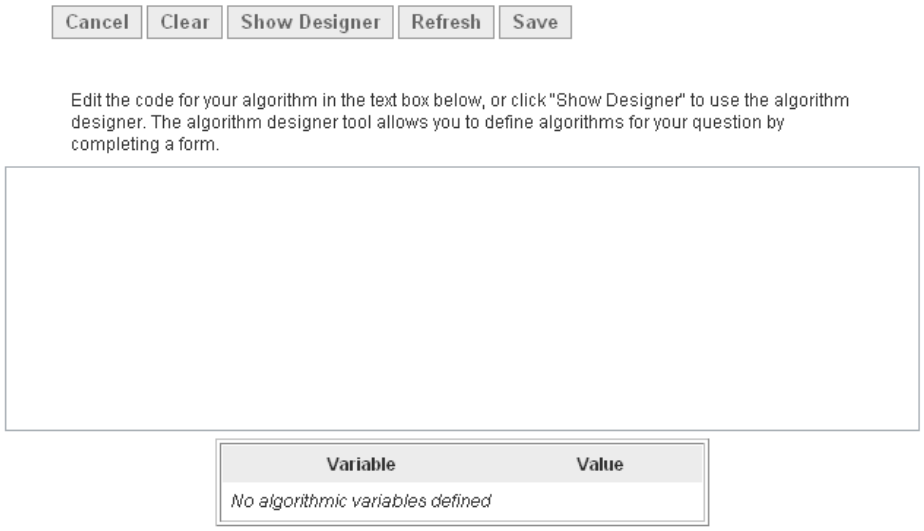

Cancel
Clear
Show Designer
Refresh
Save
Edit the code for your algorithm in the text box below, or click "Show Designer" to use the algorithm designer. The algorithm designer tool allows you to define algorithms for your question by completing a form.
Variable
Value
No algorithmic variables defined

Figure 8.1: Algorithm Editor

3. Add or edit algorithm statements (variable definitions and variable control statements).
   - If you know the system algorithm syntax, directly add or edit statements in the algorithm text region.
   - If you don't know the system syntax, click the **Show Designer** button to display the Algorithm Designer. You can only add (not edit) statements using the **Algorithm Designer**.

When you have finished adding and editing variable statements, click the **Save** button to save your changes and return to the **Question Name & Type** screen.

To add your changes to the cached question, proceed to the **Question Display** screen by clicking **Next**, and then click **Finish**.

Note: When refreshing the algorithmic variable with a plot statement, you need to change the plot statement in order for the new image to appear. Changing a **plotdevice** or **plotoption** entry will not update the plot.

See Also:

Overview of Algorithmic Question Capabilities (page 217)

Plotting Using Maple (page 201)

The Algorithm Designer

Use the **Algorithm Designer** to simplify adding algorithmic variables.

1. You can add any type of algorithmic statement. In the appropriate section of the **Algorithm Designer**, enter and select the required information in the template.
2. To add the statement and refresh the values in the algorithm text box, click **OK**.
3. Repeat the above steps to add other statements.

For detailed information, see the appropriate section:

- *Generating Random Number Variables (page 80)*
- *Generating Condition based Variables (page 80)*
- *Using Maple to Generate Random Number Variables (page 81)*
- *Establishing Conditions Between Variables (page 82)*

Generating Random Number Variables

Figure 8.2: Create a Random Number Variable

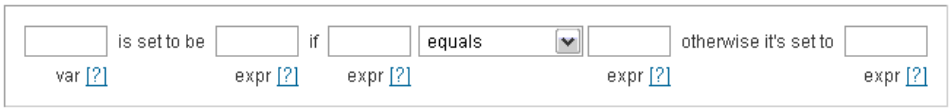
You can use the **Algorithm Designer** to create range-based variables.

1. Enter the variable name in the first cell. You do not need to use the **$** or **${}** notation.
2. Select the type of number to generate (whole number or decimal number generation with 1 to 8 decimal places).
3. Enter the minimum and maximum values for the range, and the increment (step).

> **Note:** For each expression, you can enter a:
> - Variable name, for example, **x** or **monthly totals**. You do not need to use the **$** or **${}** notation.
> - Number, for example, **5** or **2.67**
> - Math expression or formula, for example, **x^2 + 1**

4. To add the variable definition to the algorithm text box, click **OK**.
5. You can use the **Algorithm Designer** to generate multiple variables.

Generating Condition-based Variables



Create a Condition-based Variable

You can also set variable values based on conditions and relationships to other variables.

1. Enter the variable name in the first cell. You do not need to use the **$** or **${}** notation.
2. Enter the expression defining the variable if the condition *is* satisfied.
3. Enter the condition by:
   a. Entering two expressions
   b. Selecting the relation
4. Enter the expression defining the variable if the condition *is not* satisfied.

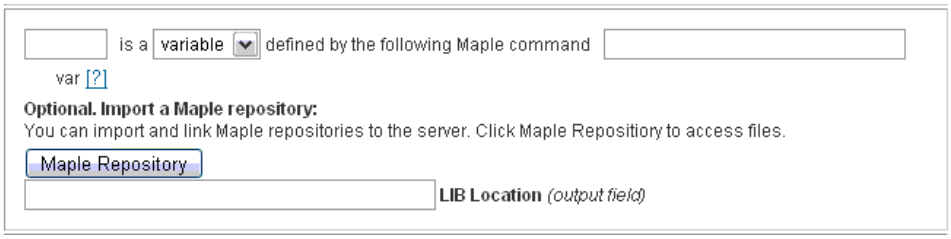
Notes:

i. Each expression you use must be a
   - Variable name, for example, **x** or **monthly totals**. You do not need to use the **$** or **${}** notation.
   - Number, for example, **5** or **2.67**
   - Math expression or formula, for example, **x^2 + 1**

ii. You must define variables referenced in the condition before defining the conditional variable.

5. To add the variable definition to the algorithm text box, click **OK**.
6. You can use the **Algorithm Designer** to generate multiple variables.

Using Maple to Generate Random Number Variables and Generate a Plot


is a variable defined by the following Maple command
var [?]
Optional. Import a Maple repository:
You can import and link Maple repositories to the server. Click Maple Repository to access files.
Maple Repository
LIB Location (output field)

Create a Random Variable Using Maple

You can use Maple commands to define random variables using *any* Maple function.

1. Start by adding the variable name in the first cell. You do not need to use the **$** or **${}** notation.
2. Select the type of variable to generate, variable or plot.
3. Enter the Maple commands used to define the random variable or enter the **plot** statement. The **plot** statement accepts plot device options. See *Setting Plot Options (page 235)* for a list of plot options.

Guidelines for Maple Commands in Maple-based Variables

i. To generate a random number using Maple, you must use the multiple commands:
randomize(): <random_command>
for **each** random variable. Otherwise, the same output is generated in each instantiation of the question.

ii. When entering multiple commands, ensure that all (except the last) have a trailing colon. The last command must *not* have a trailing colon.

iii. You *must use* the **$** or **${}** notation when including variables in a Maple command.

iv. When referencing negative, random variables in a **maple** variable definition, be sure to place the negative variable in parentheses, otherwise an error message will return.

4. To add the variable definition to the algorithm text box, click **OK**.
5. You can use the **Algorithm Designer** to generate multiple variables.

You can also use a Maple repository when generating random variables. Click the **Maple Repository** button to import and link to a Maple repository.

Establishing Conditions Between Variables

Set Conditions Between Variables

You can also use the **Algorithm Designer** to set conditions that must be met when the system generates variable values.

1. Enter the condition by:

    a. Entering two expressions

    b. Selecting the relation

> **Notes:**
> i. Each expression you use must be a:
>
> - Variable name, for example, **x** or **monthly totals**. You do not need to use the **$** or **${}** notation.
> - Number, for example, **5** or **2.67**
> - Math expression or formula, for example, **x^2 + 1**
>
> ii. You must define variables referenced in the condition before defining the condition statement.

2. To add the condition statement to the algorithm text box, click **OK**.
3. You can use the **Algorithm Designer** to define multiple condition statements.

See Also:

Overview of Algorithmic Question Capabilities (page 217)

Generating Random Numbers in Questions (page 221)

Mathematical and Logical Operators in Algorithms and Answers (page 233)

Working with Variable Data (page 218)

Adding and Editing Information Fields

When editing a question in the **Question Editor**, you can add or edit information fields. Information fields allow you to add information subfields to a question to indicate subtopic, learning objective, level of difficulty, authorship,

ownership, or other information. The information field values allow sorting and filtering in the question repository and during assignment creation, and they facilitate performance reporting by information field category.

To add or edit the info fields:

1. In the **Question Editor** screen, click **Add** or **Edit** in the **Information Fields** area.
2. Enter or edit field names and corresponding values in the **Field** and **Value** text boxes.
3. Each question can contain any number of information fields. To add rows, click the **More** button.
4. Click **Save**.
5. To add your changes to the cached question, proceed to the **Question Display** screen by clicking **Next**, and then click **Finish**.

See Also:

The info Field (page 128)

Adding and Editing Hints

When editing a question in the **Question Editor**, you can add or edit hints.

To add or edit the hints:

1. On the **Question Editor** screen, click **Add** or **Edit** in the **Hint** area.
2. Delete the text displayed and enter your comment for **hint 1** or edit the hint displayed. Hints can include text, HTML-formatted text, symbolic math expressions, algorithmic variables, inserted objects, JavaScript code, variables, and more. To add or edit HTML-formatted text, click the **Source** button in the menu bar.
3. Each question can contain any number of hints. To display a second hint, click **Enter** to display a second entry of the numeric list. Click **Enter** again to display a third hint, and so on.
4. Click **Save**.
5. To add your changes to the cached question, proceed to the **Question Display** screen by clicking **Next**, and then click **Finish**.

See Also:

Hints, Coaching, and Performance Feedback (page 128)

Adding and Editing a Solution

When editing a question in the **Question Editor**, you can add or edit the worked solution. **Note:** Information in this field is only presented to students when the question is worked in a Study Session assignment.

To add or edit the solution:

1. On the **Question Editor** screen, click **Add** or **Edit** in the **Solution** area.
2. Enter or edit the solution in the Text Editor region.
3. Click **Save**.
4. To add your changes to the cached question, proceed to the **Question Display** screen by clicking **Next**, and then click **Finish**.

See Also:

Using Complete Solutions (Study Session Assignments) (page 129)

Deleting a Comment, Information Field, Hint, or Solution

To remove a comment, hint, information field, or solution:

1. Open an existing question from the Question Repository.
2. Click the **Edit** button.
3. Click the **Edit** in the Feedback, Information Fields, Hints, or Solution area.
4. Delete the text in the entry region.
5. Click **Save**.

The system returns to the default behavior when grading the question.

See Also:

Question Fields (page 122)

Editing Question Source Files

It is **not recommended** that you use the **Edit Source** facility when editing a question until you are familiar with the source file structure.

If you are an experienced user of the system, you can use the **Edit Source** facility to edit the source for a question when editing a question in the **Question Editor**.

In this editing window, you can edit any aspect of the question script for your question, including text content for any data field.

To edit the question source in the **Question Editor**,

1. Expand a topic in the left pane and click the hyperlink of the question you want to edit.
2. Click **Edit**.
3. Click **Edit source**.
4. Edit the source as desired. When finished, click **Save**.

> **Important:** If you introduce a syntax error, for example, invalid field or value data, the question will be unusable. In particular, it is **not recommended** that you change the question type for a question.

5. You will be returned to the **Question Editor** screen. This process depends on the question type. In general, it involves clicking **Next** on the **Question Editor** screen and subsequent screens. Click **Finish** to register your changes.

See Also:

LaTeX Authoring Overview (page 247)

Using Script Files to Author Questions (page 283)

Redefining Question Types

In the **Question Editor**, when creating a new question, you can change the question type any time *before* you click **Finish**. When you click **Finish**, the system adds the question to the question repository .

To change the question type after you click **Finish**, you must edit the question source using the Edit Source facility.

Important: It is **not recommended** that you change the question type for a finished question. If you introduce a syntax error when changing the question type, the question will be unusable.

Testing the Grading of a Question

1. Preview the question by clicking the hyperlink of the question in the left pane of the **Question Editor**.
2. Enter a response to the question (as if you were a student).
3. Click **Grade**.
4. The system grades the question and displays the result.
5. Click **OK** to return to the **Display Question** screen. If there is algorithmic variable data, the random values are re-generated when the question is re-displayed.
6. To check the form of the question with different algorithmic variable data values, in the left panel, click the question name repeatedly. Each time you click the name, the question is regenerated with new random values.

Setting Answer Format in Numeric Questions

When working with **Numeric** questions in the **Question Editor**, you can specify acceptable formatting for student responses. See **Figure 8.3**.

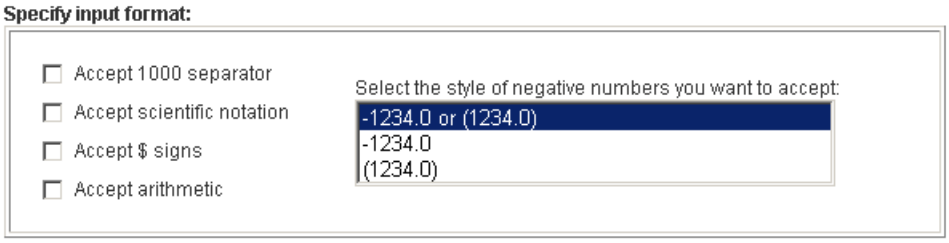


Figure 8.3: Specify Acceptable Answer Format in a Numeric Question

Accept 1000 separator - Students responses containing commas (**,**) as a separator are graded correct.

Accept scientific notation - Student responses specified using scientific notation, for example, **2.0E2**, are graded correct.

Accept $ signs - Student responses containing a leading dollar sign (**$**) are graded correct.

Accept arithmetic - Student responses specified using the arithmetic operators, that is, +, -, *, /, (), and ^, are graded correct.

Select the style of negative numbers - *Negative* responses must specified by using a negative sign (**-**), by enclosing in parentheses (**()**), or by using either notation to be graded correct. **Note:** If you select the **Accept arithmetic** check box, then you cannot allow parentheses to be used to indicate a negative response.

See Also:

Controlling Answer Format in Numeric Questions (page 182)

Numeric Questions in the Question Editor (page 102)

Setting Answer Tolerance in Numeric Questions

When working with **Numeric** questions in the **Question Editor**, you can specify the:

- Precision of student responses (that is, the number of digits)
- Acceptable margin of error for student responses
- Precision and margin of error for student responses

See **Figure 8.4**. For examples of the related plain text script files, see *Answer Tolerance Example Script (page 285)*.

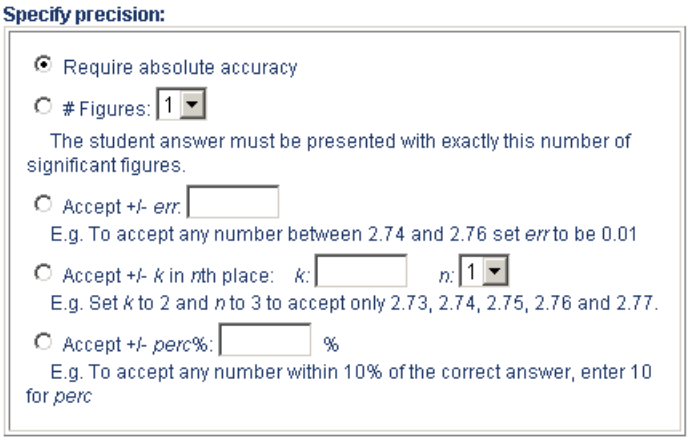

Specify precision:
Require absolute accuracy
Figures: 1
The student answer must be presented with exactly this number of significant figures.
Accept +/- err:
E.g. To accept any number between 2.74 and 2.76 set err to be 0.01
Accept +/- k in nth place: k: n: 1
E.g. Set k to 2 and n to 3 to accept only 2.73, 2.74, 2.75, 2.76 and 2.77.
Accept +/- perc%: %
E.g. To accept any number within 10% of the correct answer, enter 10 for perc

Figure 8.4: Specify Answer Precision and Tolerance in a Numeric Question

Specifying Precision

- To require that the student response *exactly* matches the *value* of the correct answer (to floating-point limits, approximately **8** decimal places), select **Require absolute accuracy**. For example, if the correct answer is **2.77**, the response **2.7701** is graded incorrect.
- To require that the student response contains a *specific number* of significant digits, and *exactly* matches the *value* of the correct answer, select **# Figures**, and then select the number of significant digits from the drop-down box. Student responses that are correct up to the specified number of significant digits receive 50% credit.

> **Important: # Figures** does **not** specify a margin of error. If you specify a correct answer value in the answer value text field with more significant digits than the number selected in the **# Figures** drop-down box, all student responses are graded incorrect. For example, if you specify **2.771** as the correct answer value, and select **3** in the **# Figures** drop-down box, **2.77** is incorrect (because it does not have the same value as the correct answer value **2.771**) and **2.771** is incorrect (because it has more than **3** significant digits).

Example

In a physics problem where the correct answer is **2.70**, if you insist that a student enter the correct number of significant digits, you must use the **# Figures** option.
If you select **Require absolute accuracy** and enter **2.70** in the answer value text field, **2.7**, **2.70**, and **2.700** are graded correct, which is not the desired behavior.
You must select **# Figures**, select **3** from the drop-down box, and enter **2.70** in the answer value text field. In this case, the student responses **2.7** and **2.700** are graded incorrect.

Setting a Margin of Error

- To directly specify the margin of error, select **Accept +/- err**, and then enter the numeric value of the variance (or corresponding algorithmic variable) in the **err** text field. For example, if the correct answer is **3.142** and **err = 0.001**, the range of correct responses is between **3.141** and **3.143** (inclusive).
- To specify the margin of error as a percentage of the correct answer, select **Accept +/- perc%**, and then enter the percentage in the **perc** text field. For example, if the correct answer is **3.141592** and **perc = 1**, the range of correct responses is between **3.11017608** and **3.17300792** (inclusive).

Specifying a Precision and a Margin of Error

- To specify the margin of error as the tolerance in the value of a significant digit and require that same number of significant digits, select **Accept +/- k in the nth place**, enter the floating-point value (or corresponding algorithmic variable) of the tolerance in the **k** text field, and then select the significant digit from the **n** drop-down box. Student responses that are correct up to the specified number of significant digits receive 50% credit. For example, if the correct answer is **2.4**, **k = 2**, and **n = 3**, the correct responses are **2.38**, **2.39**, **2.40**, **2.41**, and **2.42**. For example, if **k = 1**, **n = 3**, and the correct answer is **27.18**, the responses **27.1** and **27.2** are correct. Although **27.3** is within **0.1** of **27.18** rounded to **3** significant digits, it is incorrect because it is not within **0.1** of **27.18**.

See Also:

Setting a Margin of Error in Non-numeric Questions (page 184)

Answer Tolerance Example Script (page 285)

Controlling Answer Tolerance (page 183)

Tutorial: Setting Margin of Tolerance in Non-numeric Questions (page 215)

Images and Reference Files in Questions

You can use image files in **GIF**, **JPEG**, or **PNG** formats in questions. To place an image in a question, first upload the file to your class web site. Then include a reference to the image in the question.

You can also link to a file that has been uploaded to your class website. The following file formats are supported: htm, html, gif, jpg, png, pdf, xls, class, jar, js, css, shtml, exe, zip, swf, lib.

Uploading Images and Reference Files to the Class Web Site

1. From the **Class Homepage**, from the **Content Manager** menu, select **Web Site Editor**.
2. Navigate to the appropriate subfolder on the web site or click the **Create new subfolder** icon to create a new subfolder. For maintainability, it is recommended that you create a subfolder and place your images in it.
3. Click the **Upload file(s) to this point** icon.
4. To navigate to the file to upload, click **Browse**.
5. You can upload a single file or a zip archive.

To upload a single file:

a. Select the **Single File** radio button.

b. To specify a new name for the uploaded file on the web site, enter it in the **Save As** text field. The default name is the filename on your computer.

c. Click **OK** to upload the file.

To upload a zip archive:

a. Select the **Zip Archive** radio button.

b. To extract the files from the selected zip archive and upload them to the web site, click **OK**. (This is the recommended method for uploading multiple files.) If the file is not a zip archive, the system does not upload the file.

The system returns to the **Web Site Editor** screen. The uploaded file(s) appear in the web site explorer.

6. To upload additional files, repeat steps 3 to 5.

Important: The maximum size of a file that can be uploaded is 5 MB.

Example of Reference to Image on the Class Web Site

To include an image in a question using the Question Editor:

1. Create or edit a question in the Question Editor.
2. In the field where you want to include the image (question text, feedback, hints, or solution), click the **Insert/Edit Image** icon ().
3. Under the **Image Info** tab, click Browse Server and select the image. You will be shown a preview of the image.
4. Click **OK** to add the image.

To use an HTML reference to an image file you have uploaded to your class web site, use:

```
<img src="../classes/myclassurl/question-image.gif">
```

where **myclassurl** is the unique identifier in your class URL, and **question-image.gif** is the name of the image file.

If you have created subfolders in your class web site, use:

```
<img src="../classes/myclassurl/subfolder/question-image.gif">
```

where **subfolder** is the subfolder (or subfolders) specifying the subfolder path.

Example of Link to File on the Class Web Site

To include a link to a file (for example, a PDF) in a question using the Question Editor:

5. Create or edit a question in the Question Editor.
6. In the field where you want to include the image (question text, feedback, hints, or solution), click the **Insert/Edit Link** icon ( ).
7. Under the **Link Info** tab, click Browse Server and select the file.
8. Click **OK** to add the link.

To use an HTML reference to the file you have uploaded to your class web site, use:

```
<a href="../classes/myclassurl/filename">
```

where **myclassurl** is the unique identifier in your class URL, and **filename** is the name of the file.

If you have created subfolders in your class web site, use:

```
<a href="../classes/myclassurl/subfolder/filename">
```

where **subfolder** is the subfolder (or subfolders) specifying the subfolder path.

External Images or Reference Sources

You can use any URL that points to a resource to reference, provided the URL will be continuously available to you and your students. However, it is recommended that you use only images and references on your class web site. For external resources, enter the full URL.

Notes

- File names can consist of only alphanumeric characters and a single period.
- To include an image in a question, comment, hint, or solution, you **must** first upload the image file to the class web site on your server.
- If the image is not uploaded to the class web site, the question text appears with a **lost image** icon. You cannot check the appearance and behavior of the question properly until you upload the files to your class web site.
- You can browse and tag images in your class web site during Clickable Image question creation or editing in the **Question Editor**.

Note: If you upload a third-party question, you may need to edit the image reference URLs in the **.qu** script file before installing it. Also, you must upload any image or reference files to your class web site.

See Also:

Questions with Plots (page 148)

Flash Movies and Java Applets in Questions

You can use Flash movies or Java applets in questions. The process is similar to using an image in a question. In addition, you can create questions where students are graded directly on their interaction with the Flash move or the applet. For more information on this more advanced functionality, see *Flash Questions (page 155)*.

To include a Flash movie in a question, you must first upload the Flash movie to a subfolder on the Class web site. Next, you can include a reference to the Flash movie in your question. To do this in the Question Editor, follow these steps:

1. Create or edit a question in the Question Editor.
2. In the field where you want to include the movie (question text, feedback, hints, or solution), click the **Insert/Edit Flash** icon( ).
3. Under the Flash Info tab, click **Browse Server** and select the movie you have uploaded to the Class web site. You will be shown a preview of the movie.
4. Click **OK** to add the movie.

Graph Plotting Applet in Questions

The graph plotting applet is an interactive plot that is added to a question for dynamic plotting. The applet displays axes for the graph even when the axes lie outside of the viewable area. You can interact with the graph by shifting the viewable area or rescaling the image.

To add a graphing applet to your question:

1. Create or edit a question in the Question Editor.
2. In the field where you want to include the graph (question text, feedback, hints, or solution), click the **Source** button ( ) to edit the HTML source code directly.
3. Paste the following applet code into the source HTML where you want the graph to appear:

```
<applet code="applets.grapher.Graph" width="350" height="350"  archive="graphing.jar">
  <param name="y1" value="(x-1)(x+3)(x-5)"></param>
  <param name="xMin" value="-6"></param>
  <param name="xMax" value="6"></param>
  <param name="yMin" value="-6"></param>
  <param name="yMax" value="6"></param>
  <param name="gridLines" value="12"></param>
</applet>
```

4. Modify the applet code to reflect the actual function, plot window, and number of gridlines.

To interact with the applet:

- Click and drag the graph area to bring up a cursor that displays the coordinates of the current point
- **Shift**-click and drag the graph area to shift the viewable area
- **Ctrl**-click and drag the graph to re-scale the image

See Also:

Questions with Plots (page 148)

Plotting Using Maple (page 201)

Using a Maple Repository

You can access your personal Maple library archives from within your Maple questions. To do this, you must first upload your archives files into the class file space on the server. When you create a Maple-graded question, you then specify the location of the archive as part of the question definition. When executing the Maple code associated with that question, the Maple **libname** variable will automatically be set to access the archive files specified.

Important: When setting up multiple archives inside the same directory, ensure that there are no name conflicts. If you want to share your question bank with another class, you must share the Maple files and the test bank files.

Uploading a Maple Repository

A Maple repository consists of two files, with file extensions **.lib** and **.ind**.

To use a repository, you must upload the **.lib** and **.ind** files.

1. In the Question Repository, select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen, in the **Question Type** drop-down list, select **Maple-graded**. Click **Next**.
3. On the **Question Statement & Answer** screen, click the **Maple Repository** button beneath the **Optional. Import a Maple repository** text. The **Linking and Uploading a Repository** screen is displayed (see **Figure 8.5**).

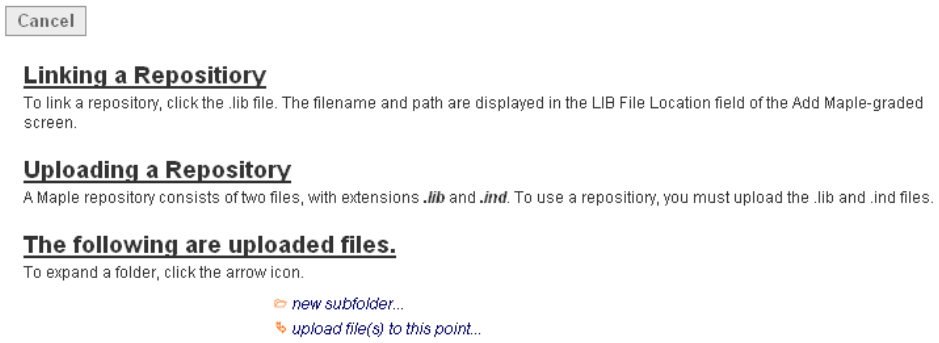
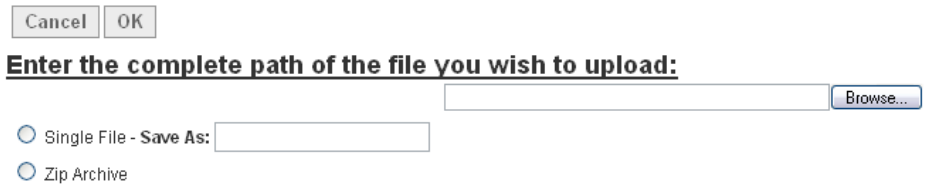

Cancel
Linking a Repository
To link a repository, click the .lib file. The filename and path are displayed in the LIB File Location field of the Add Maple-graded screen.
Uploading a Repository
A Maple repository consists of two files, with extensions .lib and .ind. To use a repository, you must upload the .lib and .ind files.
The following are uploaded files.
To expand a folder, click the arrow icon.
new subfolder...
upload file(s) to this point...

Figure 8.5: Uploading and Linking a Maple Repository

4. Select a folder (location) to which to import the Maple repository.
    - You can expand an existing folder by clicking the arrow icon.
    - To create a new folder, click the icon next to the instruction **new folder...** The folder arrow points down. The repository will upload to this folder location.
5. Click the icon next to instruction **upload files(s) to this point...** The **Upload Files** screen is displayed.



Upload Files Screen

6. Click **Browse** to locate the **.lib** and **.ind** files.
7. Select **Single File** or **Zip Archive**. You must save both the **.lib** and **.ind** files. (This requires two save procedures.)
8. Click **OK.** The Linking and Uploading a Repository opens with the new **.lib** (and subsequently) **.ind** files displayed.
9. Click the icon next to the **.lib** file. The **Question Statement & Answer** screen opens with the **.lib** path and filename in the **LIB File Location** field.

Linking to a Maple Repository

To use an existing repository, you must link to the **.lib** file.

1. In the **Question Editor/Add Mathematical Formula** screen, click the **Maple Repository** button beneath the Optional Import a Maple Repository text. The **Question Editor/Linking and Uploading a Repository** screen is displayed.
2. Expand the folders containing library files by clicking the arrow icon.
3. Click the icon next to the appropriate .**lib** file. The **Question Editor/Add Mathematical Formula** screen opens with the **.lib** path and filename in the **LIB File Location** field.
   - If you have multiple archives in the same directory, you must specify the directory name and *not* the **.lib** file. To specify the directory, edit out the **.lib** file name in the path that populates the **LIB File Location** field.

Note: Maple libraries are not available for use in algorithm variable definition statements.

See Also:

Maple-graded Questions in the Question Editor (page 96)

Maple-graded Question Type - Overview (page 140)

8.6 Question Authoring in the Question Editor

Online Authoring in the Question Editor

To access the **Question Editor**, select your class. From the **Content Manager** menu, select Question Repository. Then from the **Questions** menu, select **New Question**.

The **Question Editor** allows you to create and edit the following types of questions in an online session.

Authoring Page	**General Information on Type and Subtypes**
Clickable Image Questions in the Question Editor (page 93)	*Clickable Image Questions (page 154)*
Essay Questions in the Question Editor (page 94)	*Ungraded Essay Questions (page 155)* - not graded by the system
Free Response (Fill-in-the-Blank) Questions in the Question Editor (page 95)	*Free Response and Fill-in-the-Blank Questions (page 156)*- text field entry or drop-down box selection
Maple-graded Questions in the Question Editor (page 96)	*Maple-graded Formula (page 142)* or *Maple Syntax (page 143)*
Matching Questions in the Question Editor (page 98)	*Matching Questions (page 161)*
Mathematical Formula Questions in the Question Editor (page 98)	• *Formula with Units (page 139)*(dimensioned formula) • *Equation (page 137)* • *Formula (page 137)* • *Formula List (page 139)* • *Constants of Integration Questions (page 136)* (formula mod c) • *Multipart Formula (page 146)* • *Vectors of Formulas (page 150)* (ntuple) • *Restricted Formula (page 150)* • *Chemistry (page 135)*
Multipart Questions in the Question Editor (page 100)	*Multipart Questions (page 162)*
Multiple Choice Questions in the Question Editor (page 101)	*Multiple Choice Questions (page 163)*
Multiple Selection Questions in the Question Editor (page 101)	*Multiple Selection Questions (page 164)*

Numeric Questions in the Question Editor (page 102)	*Numeric (page 146)* (with or without units)
Palette-based Symbolic Editor Questions in the Question Editor (page 103)	*Palette Questions (page 165)*
Question Designer Questions in the Question Editor (page 104)	*Question Designer Questions (page 166)*
True-or-False Questions in the Question Editor (page 105)	*True-or-False Questions (page 177)*

The online authoring module allows you to include complex content such as tables and other display formatting by using HTML in question files.

Important:
For information about system time-outs and loss of data, see *Avoiding System Time-outs during Authoring Sessions (page 77)*. When creating and editing question banks, clicking **Finish** does not permanently save a copy of a question to a hard drive. You must save your question bank to your hard drive as a .qu plain text source file or install the question bank on the class web site hard drive.

See Also:

Creating a New Question or Editing an Existing Question (page 76)

Question Types Available in the System (page 131)

Clickable Image Questions in the Question Editor

Clickable image questions can be authored only in the **Question Editor**. The **Question Editor** provides a suite of graphics tools required to tag response areas in the source image files for correct answers.

To create a clickable image question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down box, select **Clickable imagemap**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. To proceed to the Add Clickable Imagemap screen, click **Next**.
3. On the Question Statement & Answer screen,
    a. In the text box, enter the question statement.
    b. In the **URL** text field, enter the URL where the GIF image is located. To upload or browse files you have already uploaded to your class web site:
    - Click the **Browse** button in the **Question Editor**. Follow instructions in **Uploading Images and Reference Files to the Class Web Site**.
    - Entering the path and image name populates the **Enter the URL of an Image** field.

    c. When you have located and identified the URL for your graphic image, enter the text of the question and select **Next**.
4. On the Define Correct Regions screen, use the system's graphics tools to define the regions of the correct answer and any other responses.

a. Define the approximate region of the correct answer. Select **New Region** to engage the define tool and click one corner of your answer target. Insert additional outlying points with mouse clicks, and drag the connector lines to shape the final highlighted region.

Note: The *define* tool essentially works like a peg board with rubber bands. Place the pegs with a click of the mouse, then click and drag the connecting lines while the system paints the interior of the shape you have created.

b. When you are satisfied with the shape of the region you have defined, click Set Correct to identify it as the correct response. The system highlights the correct answer region in red.

c. Use New Region to define additional regions. When you have completed each individual region, select New Region to disengage the define tool.

Note: You can use **Delete** to clear individual regions if you make a mistake.

d. When you have completed the identification of regions within your image, click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

Uploading Images and Reference Files to the Class Web Site (page 87)

Clickable Image Question Example Script (page 288)

Essay Questions in the Question Editor

To create an essay question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
   a. In the **Question Type** drop-down box, select **Essay**.
   b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
   c. You can add a comment, algorithms, information fields, hints, or a worked solution.
   d. Click **Next**.
3. On the Question Statement screen,
   a. In the text box, enter the question statement.
   b. Click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

Changing Grades and Adding Comments (page 53)

Free Response (Fill-in-the-Blank) Questions in the Question Editor

You can create two kinds of **blanks** questions by using the **Fill in the blanks** question type in the **Question Editor**.

- The **[text]** type produces a text field in the question. The student enters a response in the text box. It is graded using string matching rules. For more information, see *Rules for Grading Free Response Questions (page 158)*.
- The **[menu]** type produces a drop-down box containing the correct response and any number of incorrect responses. The student selects a response from the drop-down box.

To create a **blanks[formula]** question, use scripts or LaTeX.

To create a fill-in-the-blank question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down box, select **Fill in the blanks**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. Click **Next**.
3. On the Question Statement screen,
    a. In the text box, enter the question statement including the correct answer(s).
    b. Select the radio button (**Textboxes** or **Menus**) corresponding to the appropriate question presentation. A text box fill-in-the-blank question has a text field in which the student enters a response.
    c. Click **Next**.
4. On the Question Blanks screen,
    a. Select the words to be replaced by blanks in the question.
        - To designate a word as a blank, click the dots on either side of the word.
        - To designate a phrase (adjacent words) as a blank, click the dots on either side of the phrase.

Note: If the designated word or phrase is followed by punctuation, insert a space between the word (or last word in a phrase) and the punctuation mark.

        - To remove a blank from a word or phrase, click one of the words in the blank.
    b. For **menu** fill-in-the-blank questions, click **Add Choices**.
        i. In the text fields, enter the incorrect responses for the drop-down box (or boxes).
        ii. To add additional text fields, click **More**.
        iii. To return to the previous screen, click **OK**.
    c. Click **Finish** to place a copy in the web site cache and preview your question.

See Also:

Fill-in-the-Blank Question Example Script (page 289)

Free Response and Fill-in-the-Blank Questions (page 156)

Maple-graded Questions in the Question Editor

Creating Maple-Graded Questions

To create a Maple-graded question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down list, select **Maple-graded**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. Click **Next**.
3. On the **Question Statement & Answer** screen,
    a. Enter the text of the question. To include complicated mathematical expressions, click the Equation Editor icon ($\sum$) in the toolbar. This launches the Equation Editor.
    b. In the **Enter Maple code that evaluates to the correct answer** region,
    - The last line of code must evaluate to an expression that is stored to the variable **$ANSWER**.
    - A Maple-graded question must use valid Maple code to evaluate the answer.
    - Complete each line of code with a semicolon.
    - Add parentheses when appropriate, for example, (x + y) / 2.
    - Use the long form name for all package functions, for example, **VectorCalculus[ArcLength]**.

    c. In the **Enter Maple code to grade the student response** region,
    - A Maple-graded question must use valid Maple code to evaluate the answer.
    - The last line of your question code must evaluate to a Boolean value (true or false) or a floating-point number between 0.0 and 1.0 for partial grading. By default, use the Maple command **evalb(($ANSWER)-($RESPONSE)=0);** as indicated, where **$RESPONSE** represents the student response and **$ANSWER** represents the correct answer.
    - If you choose to alter the default grading syntax, complete each line of code with a semicolon.
    - Use the long form name for all package functions, for example, **VectorCalculus[ArcLength]**.
    - Use **$RESPONSE** to represent the student response in your code. Before your code is processed by Maple, **$RESPONSE** is replaced by the student response.
    - Use **$ANSWER** to represent the correct answer, as you have provided in the previous field.

    d. From the **Select the type of expression you want to accept** field, select **Formula** or **Maple Syntax**. To learn about the differences between Formula and Maple Syntax expression types, see *Maple-graded Question Type - Overview (page 140)*.
    e. If you selected Maple syntax, from the **Text/Symbolic entry** field, select **Text entry only** if you want students to type their response in a text region, or **Symbolic entry only** if you want students to use the Equation Editor to enter their response. For more information on this option and the advantages of each entry mode, see *Maple-graded Question Type - Overview (page 140)*.
    f. Optional. Upload a Maple repository. For instructions, see *Using a Maple Repository (page 90)*.
    g. Optional. Enter code to enable plotting of the student response. Use **$RESPONSE** to represent the student response in your code. Before your code is processed by Maple, **$RESPONSE** is replaced by the student response. The last line must be a Maple **plot** command.

h. **Note:** You can copy content from a Maple worksheet or a Maple code text file to be used in a Maple-graded question. See Copying Content from the Maple Worksheet to Maple T.A. below.

i. Click **Finish**.

For more information on Maple code, refer to your Maple system documentation.

See Also:

Editing with the Equation Editor (page 188)

Plotting a Student Response (page 187)

Applying Partial Grading to Maple-graded Questions

You can modify Maple-graded questions to allow grading between **0.0** and **1.0** for partial credit. Note: grading must be done using a floating-point number. Rational numbers are not accepted. **Table 8.1** shows a simple application of the partial credit feature.

Table 8.1: Maple-graded Question with Partial Credit

Text for the question	Solve for x in the following equation: 3x+6=12
Maple code that evaluates to the correct answer	**solve(3*x+6=12,x);**
Maple code to grade the student response	**if ($RESPONSE=6) then 0.5 else evalb($RESPONSE-$ANSWER=0) end if;**

Copying Content from the Maple Worksheet to Maple T.A.

1. From a Maple worksheet, select the code to be copied.
2. From the Maple menu, select **Copy**.
3. In Maple T.A., right-click (**Control**-click on Macintosh®) and paste the contents into the appropriate box.
4. Rename the response value to **$RESPONSE.**
5. Add **$** to all algorithmic variables.

See Also:

Creating a New Question or Editing an Existing Question (page 76)

Displaying a Maple Plot (page 201)

Using the Comment Field in a Maple Question (page 294)

Maple-graded Question Type - Overview (page 140)

Mathematical Functions & Operations (page 204)

Matching Questions in the Question Editor

The Matching question type allows only **1 to 1** matches. One to many matches are *not* available in the system with this question type.

To create a Matching question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. From the **Question Type** drop-down list, select **Matching**.
    b. In the **Question Description** field, enter an appropriate description of the matching question. To include complicated mathematical expressions, click the Equation Editor icon.
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. Click **Next**.
3. The Question Statement & Matching screen opens.
    a. In the **Text of the question** field, you can overwrite the default text with appropriate information.
    b. Enter an item to be matched in **The Item** field.
    c. Enter the matching information in the **Matches with** field.
    d. Click **Add**. The matching items are displayed in a shaded background beneath the **Display this question in __ columns** field.

Note: If you click **Finish**, the system displays the following message: *A matching question makes no sense without at least two items.* Click **OK**.

    e. Enter the number of columns in which to display the information in the **Display this question in __ columns** field.
    f. Add more matching questions by repeating steps 3b-3d.
    g. Click **Finish** to place a copy in the web site cache and preview your question.

See Also:

Matching Questions (page 161)

Question Types Available in the System (page 131)

Mathematical Formula Questions in the Question Editor

Formula questions are the fundamental question type. Student responses must contain a mathematical expression including numbers or math formulas.

To create a Formula question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down box, select **Mathematical formula**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a solution.
    d. Click **Next**.
3. On the Question Statement & Answer screen,
    a. In the question text box, enter the question statement. To include complicated mathematical expressions, click the Equation Editor icon (∑) in the toolbar. This launches the Equation Editor.
    b. In the text region labeled **Enter the correct answer:**, enter the correct answer in symbolic math. Your answer must be in the correct format. For example, for a restricted formula question, you cannot use **sin**.
    c. Select the type of expression you want to accept from the drop-down box. This determines the question type. **Table 8.2** lists the question types and corresponding expression type.

Table 8.2: Expression Types Available in Mathematical Formula Questions in the Question Editor

Question Type	Type of Expression
Formula	Formula
Restricted Formula	Formula without logs and trig
Dimensioned Formula	Formula with physical units
Formula Mod C	Formula that matches responses to within +C
Equation	Equation
Multi Formula	Unordered list of formulas
Formula List	Ordered list of formulas
Ntuple	Vector of formulas
Chemistry	Chemical equation

    d. Click **Finish** to place a copy of your question in the web site cache and preview your question.

For more information on Mathematical Formula questions, see *Math Question Types (page 132)*.

See Also:

Authoring Mathematical Questions (page 179)

Editing with the Equation Editor (page 188)

Understanding the Math Capabilities (page 181)

Multipart Questions in the Question Editor

You create a multipart question in the **Question Editor** by selecting questions for its parts from a question group.

To create and organize the parts for a multipart question:

1. Create a question group in the Question Repository that will contain the questions that will form the parts of your multipart question. Add the questions to the group.
2. If necessary, create or import questions.
3. If necessary, reorder the questions.

To create a multipart question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down box, select **Multipart question**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. Click **Next**.
3. On the **Import Questions** screen,
    a. Select the radio button corresponding to the group containing the questions to be used as parts. Click **Next**.
4. On the **Question Statement & Weighting** screen,
    a. In the text region, enter the question statement (the text to appear at the top of the question).
    b. In the **Select questions and assign weightings** box:
        i. Select the check boxes beside questions to include as parts.
        ii. Click the plus and minus buttons to assign the relative weighting of each.
    c. From the **Numbering style** drop-down menu, select the numbering style for the parts of the question.
    d. Click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

Multipart Questions (page 162)

Multipart Questions: Partial Credit (page 163)

Creating a New Question or Editing an Existing Question (page 76)

Multipart Question Example Script (page 295)

Tutorial: Working with Multipart Questions (page 112)

Multiple Choice Questions in the Question Editor

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down box, select **Multiple choice**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. Click **Next**.
3. On the Question Statement & Answer screen,
    a. In the text box, enter the question statement.
    b. In the **Choices for the answers** boxes, enter the possible choices. To add additional choice text fields, click the **Add Choices** button.
    c. If the correct answer consists of multiple choices, select the **Allow more than one selection** radio button.
    d. Select the radio button(s) corresponding to the correct answer.
    e. Click **Finish** to place a copy of your question in the web site cache and preview your question.

To add response-specific feedback for the student:

1. On the preview screen, click **Edit**.
2. On the Question Name & Type screen, click **Edit source**.
3. On the Edit Source screen, for each choice listed in the source, add a comment line of the form:
   comment.<n>=<comment_text>@
   where <n> in the comment line is the same as in the corresponding choice line:
   choice.<n>=<choice_text>@ at the end of the source code.
4. Click **Save**.
5. On the Question Name & Type screen, click **Next**.
6. On the Add Multiple Choice screen, click **Finish**.

See Also:

Multiple Choice Question Example Script (page 296)

Multiple Selection Questions in the Question Editor

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,

a. In the **Question Type** drop-down box, select **Multiple selection**.

b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)

c. You can add a comment, algorithms, information fields, hints, or a worked solution.

d. Click **Next**.

3. On the Question Statement & Answer screen,

a. In the text box, enter the question statement.

b. In the **Choices for the answers** text fields, enter the possible choices and select the radio buttons corresponding to the correct answers. To add additional choice boxes, click **Add Choices**.

c. Click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

Algorithmic Multiple Selection Question Example Script (page 284)

Multiple Selection Question Example Script (page 296)

Numeric Questions in the Question Editor

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,

a. In the **Question Type** drop-down box, select **Numeric**.

b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)

c. You can add a comment, algorithms, information fields, hints, or a worked solution.

d. To proceed to the **Add Numeric** screen, click **Next**.

3. On the Question Statement & Answer screen,

a. Enter the question statement.

b. In the **Number** text field, enter the numeric part of the correct answer in symbolic math syntax. This is required.

c. If the correct answer has physical units, in the **Units** text field, enter the units for the correct answer. To use units in addition to the default system units, you must add a custom table of units. If you do not specify units, by default, the system does not display a **Units** text field in the question.

Note: To display the **Units** text field in a question for which the correct answer has no units, edit the question source to specify **showUnits=true**. To **not** display the **Units** text field in a question for which the correct answer has units, edit the question source to specify **showUnits=false**. See *Editing Question Source Files (page 84)*

d. Specify the precision or margin of error for the grading of a student response.
Important: When creating questions that have a specific number of significant digits in the correct answer, it is recommended that you use the sig function to create algorithmic variables. By using these variables in the question statement, feedback, hints, and solution, you ensure that values are displayed with the correct number of significant digits. For more information, see *Setting Answer Tolerance in Numeric Questions (page 86)* and *decimal(n, x), sig(n, x), int(x) (page 223)*.

e. Specify the acceptable input formats for student responses. See *Setting Answer Format in Numeric Questions (page 85)*.

f. Click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

Authoring Mathematical Questions (page 179)

Using a Table of Units (page 205)

Understanding the Math Capabilities (page 181)

Numeric (page 146)

Palette-based Symbolic Editor Questions in the Question Editor

To create Palette questions that integrate your customized palette:

1. Create a question bank in plain text script format and define a palette definition(s) at the beginning of the question bank. For information on specifying a palette definition, see *Equation Editor Requirements and Syntax (page 192)*. You should also include at least one question in the question bank. (This question is simply a place-holder at this point.)
2. Import the question bank.
3. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
4. On the Question Name & Type screen,
    a. From the **Question Type** drop-down list, select **Palette-based symbolic editor**.
    b. You can add a comment, algorithms, information fields, hints, or a worked solution.
    c. Click **Next**.
5. On the Question Statement & Answer screen,
    a. Enter the text of your question statement in the **Question Text** field.
    b. Select the palette style to use from the **Palette Style** drop-down menu. (This drop-down menu lists all of the palette definitions you defined in step 1.)
    c. Enter the correct answer in the **correct answer** region.
    d. The default behavior for grading palette question student responses is to perform string-matching on the palette-created expression. For details, see *Rules for Grading Free Response Questions (page 158)*.

> **Note:** Palette questions should only be created using the **Question Editor**. Palette questions use an Equation Editor to generate Presentation MathML for the student response and compare it with the Presentation MathML of the answer. However, because there are many ways of making Presentation MathML for the same marked-up expression, creating expressions outside the Equation Editor may generate unpredictable results.

    e. Click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

Palette Questions (page 165)

Question Designer Questions in the Question Editor

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
   a. In the **Question Type** drop-down box, select **Question Designer**.
   b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
   c. You can add a comment, algorithms, information fields, hints, or a worked solution.
   d. Click **Next**.
3. On the Question Designer screen
   a. In the text box, enter the question statement.
   b. Click the **Edit/Insert Response Area** button ( ).
   c. The **Edit Response Area** dialog opens. You can select from Formula, Maple, Multiple Choice, Numeric, List, or Essay question types.
   d. Select a question type and define the properties for that question type. Click **OK**.
   e. The properties are now displayed in the **Text of the question** panel.
   f. You can add multiple parts to a question using the Question Designer by inserting additional response areas.
   g. Click **Finish** to place a copy of your question in the web site cache and preview your question.

A sample question is shown in **Figure 8.6**.

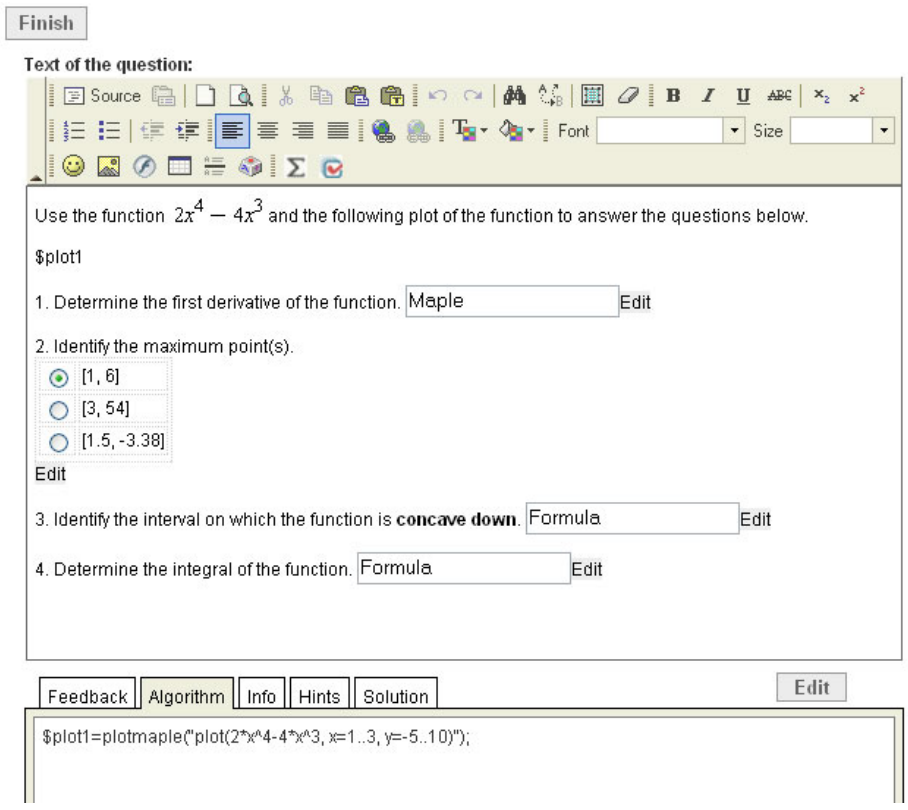


Figure 8.6: Question Designer Question

For detailed information on creating question designer questions, see *Question Designer Questions (page 166)*.

See Also:

Question Designer Question Example using Various Embedded Questions (page 298)

Algorithmic Multiple Selection Question Example Script (page 284)

Multiple Selection Question Example Script (page 296)

True-or-False Questions in the Question Editor

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down box, select **True/false**.
    b. It is recommended that you enter a description in the **Question Description** text field. (The description is used to label the question in the topic. If excluded, the question type is used.)
    c. You can add a comment, algorithms, information fields, hints, or a worked solution.
    d. Click **Next**.
3. On the Question Statement & Answer screen,
    a. In the text box, enter the question statement.
    b. Select the radio button (**True** or **False**) corresponding to the correct answer.
    c. Click **Finish** to place a copy of your question in the web site cache and preview your question.

See Also:

True-or-False Question Example Script (page 300)

Creating Pop-Up References in Questions

To add a pop-up reference to a question, insert and customize the sample HTML code (from the example below) in your question.

Example

To create a pop-up HTML reference file in a question:

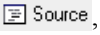
1. Create an HTML file containing instructions or reference information. For example,

```
instructions.html
```

2. Upload this file to the class web site using the **Web Site Tools** from the **System Tools Menu**.
3. Add the following sample HTML code to your question:

```
<a href="javascript:window.open('../classes/yourclassURL/instructions.html',
   'instructions',  'width=450,height=650,scrollbars').focus();">
Click here for instructions</a>
```

where **http://www.yourschool.com/classes/yourclassURL** is your class web site URL.

To add HTML code through the Question Editor, click the **Source** button, 

, in the menu bar. You can also add a pop-up reference to a question using plain text script authoring by adding this code to the **question** field (or, any HTML field in the question definition in the **.qu** script file).

Behavior

The question contains a link **Click here for instructions**. When you click the link, a new window of size 450x650 pixels opens containing the **instructions.html** file contents.

To change the description of the content in the HTML file, the filename, or the window size, edit the sample code in the example above.

Questions with Labeled Images

You can include a GIF or JPG image with labels in a question. The labels can be static or use algorithmically generated variable data.

You must first upload an image to the class web site. See *Images and Reference Files in Questions (page 87)*.

To use the Label Image applet, include the following HTML code in your question.

```
<div align = center>
    <applet code="applets.labelImage.LabelImage"  width="50" height="50">
        <param name="image" value="../classes/class1/Image1.gif">
        <param name="size" value="1">
        <param name="label.1.x" value="20">
        <param name="label.1.y" value="20">
        <param name="label.1.text" value="text for label">
    </applet>
</div>
```

where

- The **div** tags are optional.
- The **width** and **height** tags specify the dimensions of the image in pixels. To determine the width and height of the graphic, use a third-party graphics application (for example, Microsoft® Paint).
- The **image** parameter **value** specifies the path and filename of the image file.
- The **size** parameter **value** specifies the number of labels. You can specify zero or more labels.

The location and contents of each label are specified using three parameters.

1. The **label.i.x** parameter **value** specifies the horizontal offset (in pixels) of the middle of the i^{th} label from the left side of the image.
2. The **label.i.y** parameter **value** specifies the vertical offset (in pixels) of the baseline of the i^{th} label from the top of the image.
3. The **label.i.text** parameter **value** specifies the content of the i^{th} label. It can include algorithmic variables.

To add HTML code through the Question Editor, click the **Source** button, Source, in the menu bar. You can also add a Label Image applet to a question using plain text script authoring by adding this code to the **question** field (or, any HTML field in the question definition in the **.qu** script file).

See Also:

Images and Reference Files in Questions (page 87)

Questions with Plots (page 148)

Overview of Algorithmic Question Capabilities (page 217)

Labeling Graphics in LaTeX Questions (page 263)

Labeling Graphics in Questions Example Script (page 291)

Text Formatting in Questions

You can use HTML 4.0 tags to control the display of character formatting in most question fields. (For a list, see *Special Characters in Questions (page 130)*.) To do so, click the **Source** button, Source, in the tool bar of the text editor region of the question and include the appropriate format tags.

Note: This requires an HTML or text editor and knowledge of HTML.

Table 8.3: Examples of HTML Tags to Use for Formatting Text

HTML Formatting	Function
`<p>Text of paragraph</p>`	Sets the enclosed text as a paragraph
` Text of new line`	Sets a new line break (carriage return) in the current paragraph
`<b>Text</b>`	Boldfaces the enclosed text
`<i>Text</i>`	Italicizes the enclosed text
`<sub>TexFot</sub>`	Subscripts the enclosed text
`<sup>Text</sup>`	Superscripts the enclosed text
`<font face="..">Text</font>`	Sets text font
`<font size="..">Text</font>`	Sets text size
`<font color="..">Text</font>`	Sets text color

Tutorial: Maple Syntax Questions with Matrices

A Simple Matrix Question

This tutorial describes how to create a question that asks for the coefficient matrix of a linear system of equations. Students will be able to use the Equation Editor to construct their response.

This question is created in the Question Editor as a Maple-graded question.

To create the question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down list, select **Maple-graded**.
    b. Enter the description **Coefficient Matrix**.
    c. Click **Add** in the Algorithm area.
3. In the Algorithm screen,
    a. Define a random variable:

```
$ans=maple("Matrix([[1,2,-3],[7,-1,2]])");
```

    b. Click **Refresh** to preview the variable you defined. See **Figure 8.7**.
    c. Click **Save**.

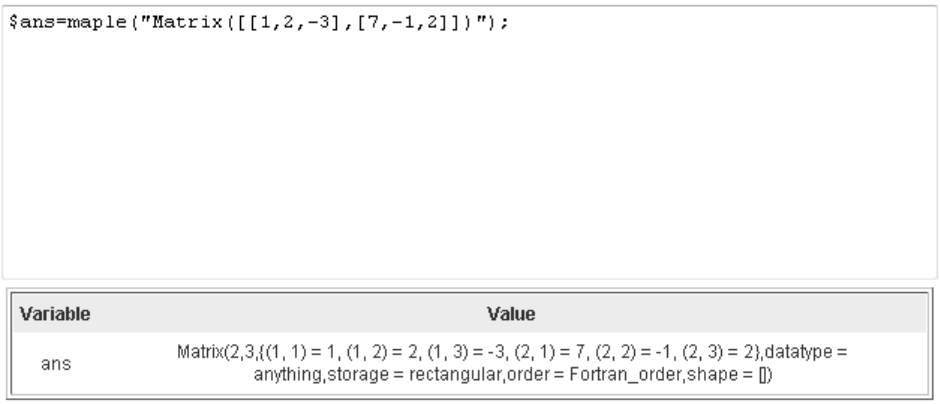

$ans=maple("Matrix([[1,2,-3],[7,-1,2]])");
Variable
Value
ans
Matrix(2,3,{(1, 1) = 1, (1, 2) = 2, (1, 3) = -3, (2, 1) = 7, (2, 2) = -1, (2, 3) = 2},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])

Figure 8.7: Defining a Matrix

4. Click **Next**.
5. On the **Maple-graded Question and Answer screen,**

    a. Enter the text of the question:

    Give the coefficient matrix of the following system.

    $$x + 2y - 3z = 1$$

    $$7x - y + 2z = 4$$

    To enter the system of equations, click the Equation Editor icon ($\sum$) in the toolbar, and enter the expressions in the Equation Editor.

    b. In the **Enter Maple code that evaluates to the correct answer** region, enter

    ```
    printf(MathML[ExportPresentation]($ans));
    ```

    Note: This generates MathML for the matrix, presenting it in a nice format for the students. For the purpose of grading the student response, however, you must use the **$ans** variable, not the MathML expression.

    c. In the **Enter Maple code to grade the student response** region, enter

    ```
    LinearAlgebra[Equal]($RESPONSE,$ans);
    ```

    Note: The default grading code, **evalb(($ans)-($RESPONSE)=0);**, is not sufficient when working with matrices. To check for matrix equality, use the command **LinearAlgebra[Equal]**.

    d. From the **Select the type of expression you want to accept** field, select **Maple Syntax** because students will be entering a matrix.

    e. From the **Text/Symbolic entry** field, select **Symbolic entry only** so that students can use the Equation Editor to create the matrix rather than having to enter the Matrix command. See **Figure 8.8**.

    f. Click **Finish** to save the question.

    g. Add the question to one or more question groups, and then click **Submit**.

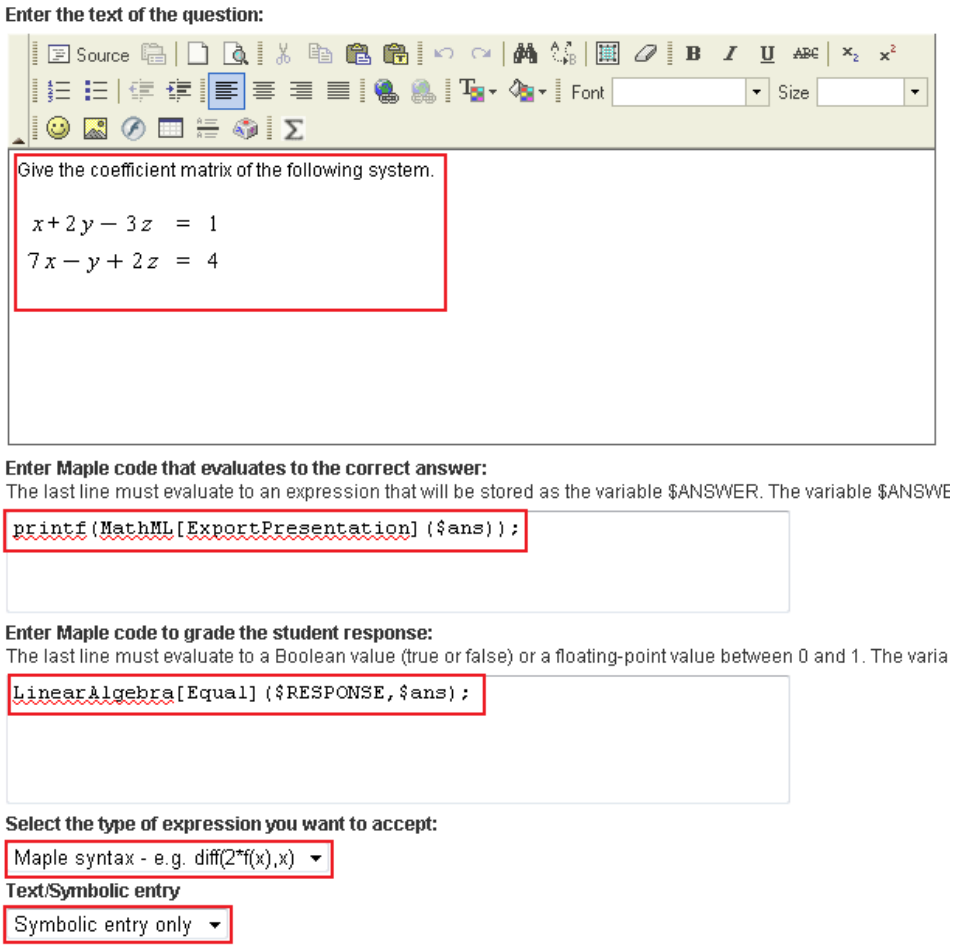

Enter the text of the question:
Source
Font
Size
Give the coefficient matrix of the following system.
$x+2y-3z = 1$
$7x-y+2z = 4$
Enter Maple code that evaluates to the correct answer:
The last line must evaluate to an expression that will be stored as the variable $ANSWER. The variable $ANSWE
printf(MathML[ExportPresentation]($ans));
Enter Maple code to grade the student response:
The last line must evaluate to a Boolean value (true or false) or a floating-point value between 0 and 1. The varia
LinearAlgebra[Equal]($RESPONSE,$ans);
Select the type of expression you want to accept:
Maple syntax - e.g. diff(2*f(x),x)
Text/Symbolic entry
Symbolic entry only

Figure 8.8: Defining a Simple Matrix Question

To answer this question, students interact with the Equation Editor. A student can right-click (**Control**-click on Macintosh) in the Equation Editor to open the palettes. From the Matrix palette, the student must select the appropriate size of matrix (see **Figure 8.9**). Then, they must fill in the values for the matrix entries.

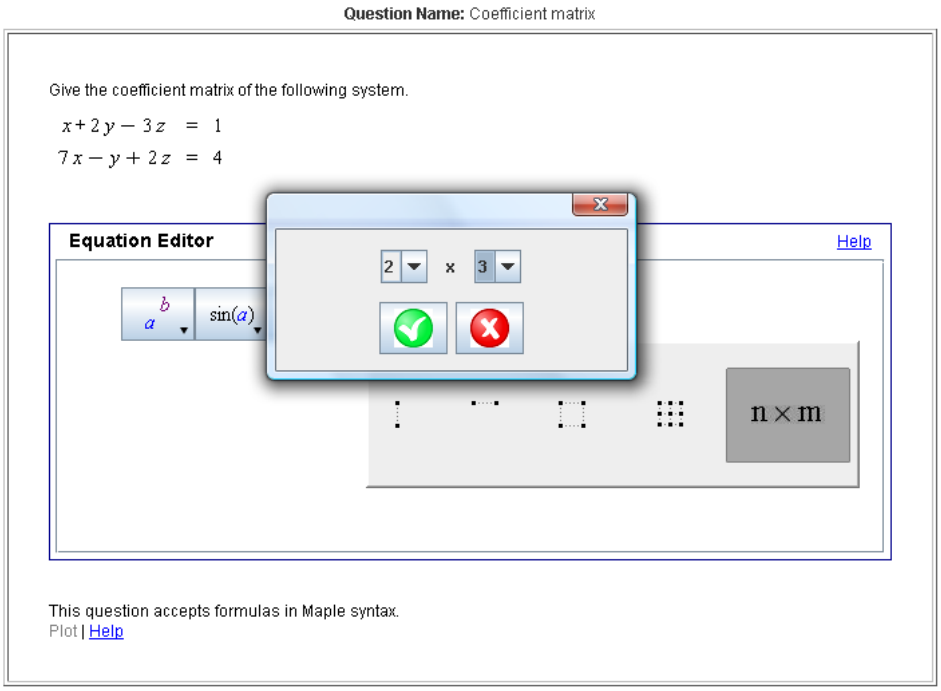


Figure 8.9: Using the Matrix Palette

Note: One of the skills tested with this question is determining the appropriate matrix dimensions. With the more limited Matrix question type, the size of matrix is already fixed, and the student simply enters the matrix entries.

See Also:

Maple-graded Questions in the Question Editor (page 96)

Maple-graded Question Type - Overview (page 140)

Editing with the Equation Editor (page 188)

Matrix Question with Infinitely Many Correct Answers

This tutorial describes how to create a question that asks for an invertible matrix. The grading of the student response checks whether the determinant of the student's matrix is non-zero.

This question is created in the Question Editor as a Maple-graded question.

To create the question:

1. Select the **Questions** menu, then **New Question**. The Question Editor is displayed.
2. On the Question Name & Type screen,
    a. In the **Question Type** drop-down list, select **Maple-graded**.
    b. Enter the description **Invertible Matrix**.
    c. Click **Next**.
3. On the **Maple-graded Question and Answer screen,**
    a. Enter the text of the question:

    Give an invertible matrix.

    b. In the **Enter Maple code that evaluates to the correct answer** region, enter

```
M:=Matrix([[1,2],[3,4]]):
printf(MathML[ExportPresentation](M));
```

    c. In the **Enter Maple code to grade the student response** region, enter

```
x:=LinearAlgebra[Determinant]($RESPONSE):
evalb(type(x,numeric) and x<>0);
```

    d. From the **Select the type of expression you want to accept** field, select **Maple Syntax** because students will be entering a matrix.
    e. From the **Text/Symbolic entry** field, select **Symbolic entry only** so that students can use the Equation Editor to create the matrix rather than having to enter the Matrix command. (See **Figure 8.10**).
    f. Click **Finish** to save the question.
    g. Add the question to one or more question groups, and then click **Submit**.

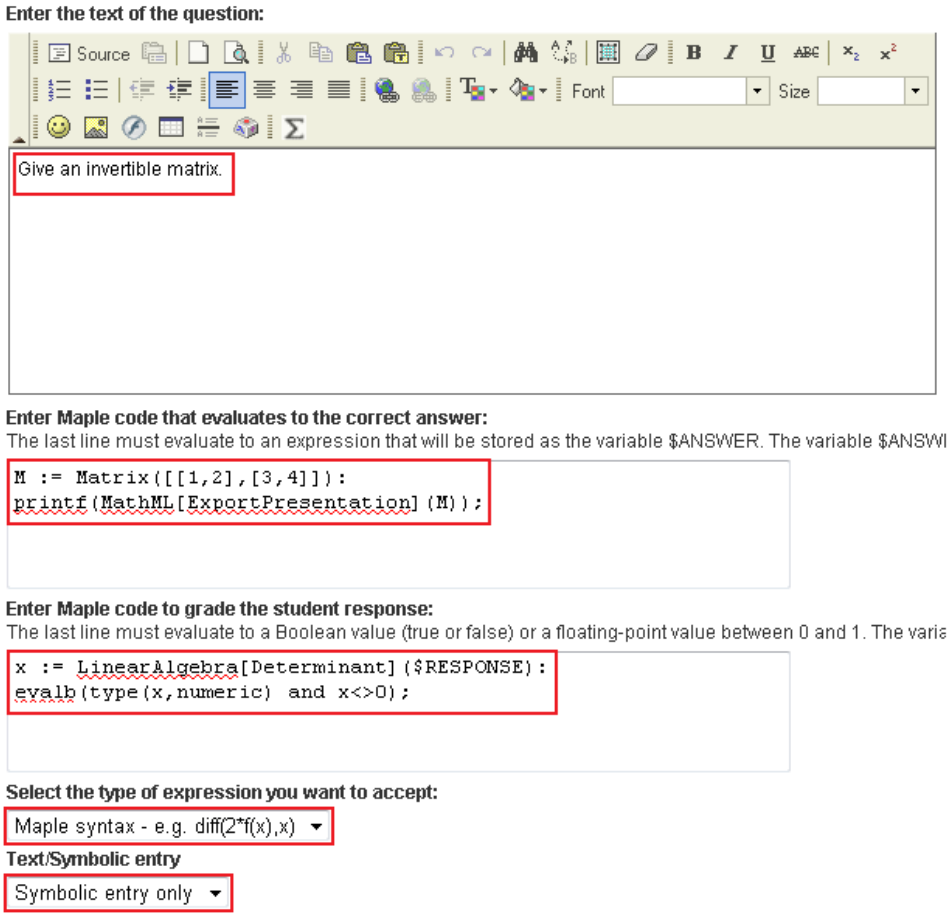

Enter the text of the question:
Source
Font
Size
Give an invertible matrix.
Enter Maple code that evaluates to the correct answer:
The last line must evaluate to an expression that will be stored as the variable $ANSWER. The variable $ANSWI
M := Matrix([[1,2],[3,4]]):
printf(MathML[ExportPresentation](M));
Enter Maple code to grade the student response:
The last line must evaluate to a Boolean value (true or false) or a floating-point value between 0 and 1. The varia
x := LinearAlgebra[Determinant]($RESPONSE):
evalb(type(x,numeric) and x<>0);
Select the type of expression you want to accept:
Maple syntax - e.g. diff(2*f(x),x)
Text/Symbolic entry
Symbolic entry only

Figure 8.10: Invertible Matrix Question

For more information on Maple code, refer to your Maple system documentation.

Tutorial: Working with Multipart Questions

This tutorial describes:

- *Editing General Information (page 113)*
- *Adding and Editing Individual Question Parts of a Multipart Question (page 113)*
- *Partial Credit and Weighted Scoring (page 113)*

For an introduction to multipart questions, see *Multipart Questions (page 162)*.

For instructions on creating a new multipart question, see *Multipart Questions in the Question Editor (page 100)*.

Editing the Top Level of a Multipart Question

1. In the **Question Editor**, select the multipart question to edit. Click **Edit**.
2. On the Question Name & Type screen, edit the description, solution, or other options. Click **Next**.
3. On the **Question Statement & Weighting** screen, you can:
    - Edit the text to appear at the top of the question
    - Specify preferences for numbering the parts using the **Numbering style** drop-down menu.
    - Set question weighting for the parts
4. Click **Finish**.

Adding to and Editing Individual Question Parts of a Multipart Question

If your question includes variables, you *must* edit the plain text script file for it to preserve your variable definitions. You can edit the script file by using the **Question Editor Edit Source** facility.

To edit using **Edit Source**:

1. In the **Question Editor**, select the multipart question to edit. Click **Edit**.
2. On the Question Name & Type screen, click **Edit Source**.
3. Edit the file.
4. To save your changes, click **OK**.
5. On the Question Name & Type screen, click **Next**.
6. On the **Question Statement & Weighting** screen, click **Finish**.

Setting Partial Credit and Question Weighting in a Multipart Question

You can set the weighted scoring of individual question parts in a multipart question.

To set the weighting for each component part of a multipart question:

1. In the **Question Editor**, select the multipart question to edit. Click **Edit**.
2. On the Question Name & Type screen, click **Next**.
3. On the **Question Statement & Weighting** screen, update the weighting. The part weighting is relative. The default setting is to equally weight each part. That is, if there are **x** parts in the question, each part is weighted **1/x** of the total value assigned to the question in an assignment. To change the relative weighting of question parts, use the plus and minus buttons in the **Weighting** column.
4. Click **Finish**.

Example

Assume you have a multipart question with 3 parts. Initially, each part is assigned a weight of 1, which indicates that 1/3 of the total value is assigned to each question. (You determine the total question value during assignment definition.) To update the weighting of the first part of the question to 5 times that of the other two parts, set the weightings to:

Part 1 - 5 (that is, 5/7 of the total question value)

Part 2 - 1 (that is, 1/7 of the total question value)

Part 3 - 1 (that is, 1/7 of the total question value)

If you add the question to an assignment, and set its value in the assignment as **10 points**, the value for the question parts would be:

Part 1: 5/7 x 10 total points = 7.14 points

Part 2: 1/7 x 10 total points = 1.43 points

Part 3: 1/7 x 10 total points = 1.43 points

Total = 10.00 points

See Also:

Multipart Questions in the Question Editor (page 100)

Multipart Questions - Sequencing Parts (page 163)

8.7 Question Repository

To access the question repository, from the Class Homepage, select **Content Manager>Question Repository**.

The question repository is used to create, organize, and search for questions.

The default view of the question repository shows your private questions, organized by groups and subgroups. You can also search public (shared) questions, either from a specific class or by all authors.

Searching for Questions in the Question Repository

You can search for questions by group, question type, keyword, or info fields.

- group — view questions within a particular *question group* in the question repository
- question type — find questions by *question type*
- keyword — search all questions for any question that contains a keyword within the *text of the question*. This is a case-sensitive search.
- info field — search the *information fields* of questions

Some questions may not belong to any group. To search for these questions, select **Questions not in groups**.

You can also view your shadowed questions and inherited questions.

To perform a search, select the desired search criteria. Then click **Search**. The search results are shown in the right panel of the question repository screen. See **Figure 8.11**.

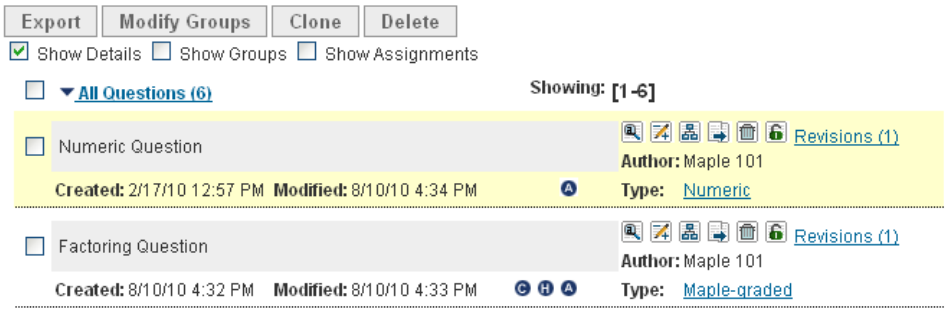


Figure 8.11: Search Results in the Question Repository

In the search results, you can view summary information for each question and sort questions based on question type, creation date, and date last modified. You can change the view options (**Show Details**, **Show Groups**, **Show Assignments**) to show more or less information about each question.

A revision history is kept for each question. You can access earlier versions of a question by clicking on **Revisions**.

Table 8.4 lists buttons may be available for each question.

Table 8.4: Question Repository: Available Actions

Icon	Description
	preview the question or assignment
	edit the question
	modify the groups to which the question belongs
	clone the question (creates a copy of a question)
	delete the question
/	share the question (make public) / make private

Not all these options are available for all questions. Inherited and shadowed questions cannot be edited, deleted, or shared. However, you can clone an inherited question to create a local, editable copy of the question.

You can also perform operations on a set of selected questions using the buttons at the top of the search results. The operations are **Export**, **Modify Groups**, **Clone**, and **Delete**. Export saves the selected questions to a question bank file (.**qu** file).

Organizing Questions in the Question Repository

In the question repository, questions are organized into groups. You can add a new group, rename a group, add a child group, or delete a group.

Note: If you delete a group, the questions in that group are no longer associated with a group. To find such questions, select **Questions not in groups**.

Deleting Questions from the Question Repository

In the question repository, you can delete a question.

To delete a question, select the question and click **Delete**. (Alternatively, click the icon beside your question.)

To view the deleted questions, in the Question Repository, select **My Deleted questions**. In addition to the standard actions, there are these actions:

- Undelete
- Permanently Delete
- Cleanup

Undelete restores the selected deleted questions. The questions are restored to their original groups.

Permanently Delete removes the selected questions shown on the current page of the search results.

Note: A question that is referenced by an assignment or by test results will not be removed. You must use the **Assignment Editor** to delete the reference to the question (see *Assignment Editor Step 2: Select Questions (page 28)*), and then save the assignment (see *Assignment Editor Step 4: Review & Finish (page 32)*).

Cleanup performs a full cleanup of the question repository for the current class. The Cleanup action will:

- permanently delete all deleted questions
- permanently delete all unused old revisions of all questions (including earlier versions of active questions)

Note: A question that is referenced by an assignment or by test results will not be removed.

Previewing a Question

1. In the left frame of the **Question Repository**, search for a question, group, or type of question.
2. In the right frame, in the search results, click the **preview** icon ( ) beside the question.

In **Preview Mode**, you can interact with the question as a student, for example, enter a response to the question in the input cells or use check boxes and radio buttons. To test the grading of the question, select **Grade**.

If the **Show Assignments** check box is selected, you can also preview the entire assignment that contains the question. To preview the assignment, click the **preview** icon beside the assignment.

Exporting Work to Your Hard Drive

You can back up questions by downloading them to your hard drive. The questions are saved as a question file (a plain text file which ends in a **.qu** extension).

To export a copy of your work on the local hard drive on your computer:

1. From the **Question Repository**, find the questions you wish to save.
2. Select the check boxes beside the questions you wish to export.
3. Click **Export**. The files are saved as **questions.qu**.

Why Export Your Questions?

You may want to have your own local copy of your questions (which normally reside on the Maple T.A. server). You may also want to share your content with others who are not part of your institution. If you save your questions to a

question file (**.qu** file) or to a course module and share that file with another instructor, that instructor can import the .qu file or course module into the Question Repository for their class in their instance of Maple T.A.

Note: If you make changes to the **.qu** file (only recommended if you are familiar with the syntax and structure of these files), you can update the questions in your own class by importing them back into the Question Repository.

8.8 Sharing Questions

Sharing content makes a wider range of questions available to all users. You can share questions from your class with other instructors in the system. You can also obtain questions from other instructors.

Questions belong to a class. The creator of a question becomes the owner of the question. Questions can only be modified by the owner.

Instructors can obtain questions in three ways:

4. *Inherit* a question from a parent class. Child classes automatically inherit questions from their parent class. Instructors can use an inherited question in their assignments, but cannot modify the question. Only the original owner (creator) of the question can modify the question, and those changes will be automatically propagated to all subscribers' question groups. This is similar to the Question Bank inheritance rules in Maple T.A. 4.0.
5. *Shadow* a public question from another class.
   Instructors can use a shadowed question in their assignments, but cannot modify the question. Only the original owner (creator) of the question can modify the question, and those changes will be automatically propagated to all subscribers' question groups.
6. *Clone* a question (public or private) from the question repository. This operation creates a local copy, or clone, of the original question and sets the owner to the instructor who made the clone.
   Instructors who clone a question can edit the cloned question. (No one other than the instructor who created the cloned question will be able to modify the clone.) Changes made to the original question will not be visible in cloned questions.

When you inherit, shadow, or clone a question, it is available in the question repository for your class, and you can use it in assignments.

To determine which method of obtaining questions is appropriate for your class, take the following behaviors into consideration and choose the method that makes the most sense for you:

- When an instructor inherits or shadows a public question, the instructor should expect it to be modified by the question owner at any time without notice. However, the question is likely to be a valid question with all resources (images, etc.) in place.
- When an instructor clones a question, the instructor becomes the full owner of the clone, and can be certain that it will not be edited by anyone else. However, there is a danger of losing resources (images, etc.) because the cloning process does not copy the resource files to the class web site; therefore, it is possible that the resources files may be deleted by the original author at any time.

To share a question:

1. In the Question Repository, find the question you wish to share.
2. Click the share icon ( ) , located beside the question in the right panel of the question repository screen.

This method works well when you are sharing a limited number of questions. Other methods for sharing content include using question banks and course modules and making your class a feature class.

To shadow a question:

1. In the Question Repository, select **Public Questions**.
2. Use the search methods to find the public questions you wish to use.
3. Add the question to one or more of your question groups by clicking the modify icon ( ) beside the question. Or, select the check boxes beside the questions you wish to use, and click **Add to Groups**. Select one or more question groups. By adding a public question to one of your question groups, you shadow the question.

To clone a question:

1. In the Question Repository, find the questions (Public or Private) that you wish to clone.
2. Click the clone icon ( ) located beside a question. Or, select the check box beside the questions you wish to clone, and click **Clone**.
   Note: Newly cloned questions may not belong to a group. In that case, you can find these questions by selecting **Questions not in groups**. You can then add them to a group.

Inherited questions are not obtained individually. A child class inherits content from the parent class. Thus, in a child class, inherited questions automatically appear in your question repository. For more information, see *Shared Classes Overview (page 13)*.

See Also:

Course Modules (page 61)

Creating a Shared Class (page 14)

8.9 Question Banks

What is a Question Bank?

Question banks are plain text files, with a **.qu** extension, which store sets of Maple T.A. questions. Question banks are organized into topics, which contain individual questions. Use topics to organize questions, for example, to match the structure of your course. Topics can also indicate learning objectives, key topics, or other organizational schemes.

Using the questions in question banks, you can create assignments.

Topic Structure within Question Banks

Each question bank consists of topics, with questions organized by topic. Questions consist of a series of related statements, all containing the same prefix indicating the topic and question numbers.

- Topics must be numbered in ascending order, starting at 1.
- Questions and topics must be numbered consecutively.
- Each topic has a name, which is a text string set using the line:

  ```
  qu.x.topic=<topic_name>@
  ```

- **x** is the topic number

The following example shows a question bank with two topics. There are two questions in the first topic, and only one in the second topic. In general, question banks contain many questions in each topic.

Example

```
qu.1.topic=Differentiating polynomials@
qu.1.1.mode=formula@
qu.1.1.question=What is the derivative of x<SUP>2</SUP>+3x+1?@
qu.1.1.answer=2x+3@
qu.1.2.mode=formula@
qu.1.2.question=What is the derivative of x<SUP>3</SUP>+x+1?@
qu.1.2.answer=3x<SUP>2</SUP>+1@
qu.2.topic=Differentiating exponentials@
qu.2.1.mode=formula@
qu.2.1.question=What is the derivative of e<SUP>x</SUP>?@
qu.2.1.answer=e<SUP>x</SUP>@
```

Question File Structure Details

Questions are stored with embedded HTML and MathML tags, in a text file called a *question file* (**.qu** plain text script file) that consists of a list of topics. In each topic is a list of questions.

Each question file contains one or more topics. Each topic has a corresponding topic statement in the script file.

qu.x.topic=<topic_name>@

- **x** is the topic number

The other statements in a question script file define questions. Each question consists of a series of related statements, all containing the same prefix, which indicates the topic and question numbers. They are of the form:

qu.x.y.<field_name>=<data>@

- **x** is the topic number
- **y** is the number of the question within the topic
- Each question has a number of fields that specify, for example, the text of the question, the correct answer, additional comments displayed when graded, and hints. Some fields are required, others are optional.
- Question fields are populated with data that can span several lines of text.
- The end of each data field statement must be marked with an **@** character.
- Questions and topics must be numbered consecutively and arranged in increasing order. That is, **x** starts at **1** and **y** starts at **1** for each new topic.

See Also:

Question Fields (page 122)

Question Modes (Question Types) (page 124)

Opening a Saved Question Bank File

While editing a question bank, it is recommended that you back up your work by exporting it to your hard drive. You can upload the saved question bank file (plain text script file), and then continue editing or save it in your class web site.

To open a question bank file that is saved on your hard drive:

1. In the **Question Repository** page, select the **Questions** menu, then **Import Question Bank**.
2. Click the **Browse** button and navigate to the source file on your hard drive. Click **Open**.
3. Select how the questions will be organized when importing.
4. Click **Import** to import the question bank. The topics are displayed in the left panel of the Question Editor.
5. Save your imported question bank. From the **Actions** menu, select **Save Question Bank**.

When you upload a question bank file into the system, it performs a set of validation routines to ensure that the questions function. Occasionally, you may encounter an error with a variable name or definition upon uploading. The following is a summary of the system's analysis of variables upon initial loading of a question bank file.

- For variables displayed within HTML text, for example, in a question statement or feedback area, a problematic variable does not produce a data value generated from its definition. Instead its variable name is displayed, highlighted in red.
- For variables appearing in non-displayed field parameters, for example, the answer to a question, the variables **$ANSWER** and **$RESPONSE** default to 0. All others have the variable name passed.

Saving a Question Bank

When working in the **Question Editor**, the system does not save the changes to the class web site until you click **Save Question Bank** from the **Actions** menu. After you install a question bank, it is available for assignment creation.

If you are re-saving an updated version of a question bank, you can:

- Replace the previous version by using the same name. Assignments that reference the previous version are automatically edited to use the updated questions from the new version.
- Use a new name. Assignments that reference the previous version continue to use the previous versions of the questions. To use the new questions, create new assignments or edit old assignments to reference the updated question bank.

To save a question bank that has been opened and edited, uploaded, or created:

1. If necessary, return to the **Question Bank Editor** main menu.
2. Select **Save Question Bank** from the **Actions** menu.
3. Enter a question bank name. (If this is an updated version of an existing question bank, consider the implications outlined above.)
4. If you are replacing a previously installed question bank, the system displays the following message.

> Warning: You are about to overwrite the question bank "Text Bank 3". This will replace all the topics and questions in that question bank with your current topics and questions. **Note:** If you save your question bank under this title, any assignments that you have set up to use the question bank that's currently saved with this title will use the new question bank from now on. However, records of assignments that students have already taken will continue to show the questions from the old question bank. Click "OK" to continue and overwrite the existing question bank, or "Cancel" to pick a different title.

To replace the superseded question bank with your new version, click **OK**.

Note: Saving a modified Question Bank with the same name as an inherited Question Bank from a shared parent class severs the relationship with the parent class version of the question bank. As a result, you will not automatically receive any changes made to the parent class version of the Question Bank.

5. The system reports "**Installation successfully completed**".

Saving to a File versus Reinstalling a Question Bank

Do not use Save Question Bank from the **Actions** menu to back up your question bank as you edit it. It is recommended that you use **Export Question Bank** from the **Actions** menu to make backups and use **Save Question Bank** only after you are finished editing a question bank.

If you use **Save Question Bank** to make backups, there are side effects:

- The backup is used in assignments, including tests.
- The backup is saved as an extra file on the server. Every time you use **Save Question Bank**, the system creates an additional copy of the question bank file on the server with a unique version number. In general, files are small (less than 100 KB), but if you frequently make backups (as is recommended), there will be many unused versions on the server. Because the system administrator cannot determine which versions have been accessed in student assignments, it is difficult to delete unused files.

Creating and Opening Modules

If you want to share not just your questions, but also images and assignments that you have created based on those questions, you should use course modules. Once you create a course module, you can send it to other instructors who can upload it into their classes and use it as is or alter it to better suit their needs. Course modules can be uploaded to other installations of Maple T.A. not just the one at your institution.

To create a course module:

1. Log in as an instructor and click on the class that contains the content you want to share.
2. Click **Content Manager**, then **Course Modules**.
3. Click **New**.
4. Select the content that you want included in the course module. Click **OK.**
5. Enter a name for the module. Optionally, enter a description for any of the items you have included in the module. Click **OK**. The course module now appears in the list of course modules for your class.

To open a module, you need to import it and install it.

To import a course module:

1. Log in as an instructor and click on the class in which you want to import the content.
2. Click **Content Manager**, then **Course Modules**.
3. Click **Import**.
4. Click **Browse...** to locate the course module file (.zip) on your computer.
5. Click **Import**. The system will display the content from the course module. If there are items that use the same name as those in your class, they will be highlighted in yellow on the upload screen.
6. Select the check boxes beside the items you want to install and click **Install**. The selected items are installed in your class.

Question Fields

Each question *must* contain **mode**, **question**, and **answer** statements. In addition, there are many optional fields that can be added to questions to enhance functionality, add comments or hints, or otherwise define or expand the capabilities of the question. **Table 8.5** lists the available question fields and summarizes related information.

Table 8.5: Question Fields

qu.x.y. <field name>	**Required or optional**	**Can include HTML or MathML?**	**Field Type**	**Description / Function**
mode	required	No	internal directive	System command that identifies the question type
type	required (in Maple-graded questions)	No	internal directive	Specifies subtype of Maple question. For Maple Syntax questions, it is **maple**. For Maple Formula questions, it is **formula**.
allow2d	optional (in Maple-graded questions)	No	internal directive	For Maple-graded questions, specifies the student entry mode. Can be 0, 1, or 2. For **maple** questions, it can be 0 (for text-entry mode) or 1 (for symbol mode). For **formula** questions, it must be 1.
name	optional	Yes	display - no variables	Displays text that identifies the individual question.
editing	optional	No	internal directive	Indicates use of HTML editing in the question. Set directly or automatically by **Question Editor**.
question	required	Yes	display - can contain variables	The question statement. Can contain any combination of text, MathML, HTML, tables, or references to images or objects.
algorithm	optional	No	internal directive	Contains variable definitions for use in the question. Must conform to the system formula and variable syntax.
answer	required (except in essay, matching, and Maple-graded questions)	No	internal directive	Defines correct answer for use in grading. If mathematical question, must conform to the system math/formula syntax.

qu.x.y. **<field name>**	**Required or optional**	**Can include HTML or MathML?**	**Field Type**	**Description / Function**
answer.number answer.unit	required (in numeric questions)	No	internal directive	Defines correct number and units of answers in numeric questions.
maple	required (in Maple-graded questions)	No	internal directive	Maple code to grade student responses and prevent cheating
plot	optional	No	internal directive	Maple plot command that plots a student response or related function. Available in all Maple-graded questions.
grading	optional	No	internal directive	Defines grading mode in numeric questions. Required in numeric questions.
units	optional	No	internal directive	Reference to a customized table of units
digit	optional	No	internal directive	Number of significant digits. Required for **exact_sigd** and **toler_sigd** grading modes in numeric questions.
perc	optional	No	internal directive	Percentage error (or tolerance). Required for **toler_perc** grading mode in numeric questions.
err	optional	No	internal directive	Absolute error (or tolerance). Required for **toler_sigd** and **toler_abs** grading modes in numeric questions.
tolerance	optional	No	internal directive	Absolute tolerance for formula questions (including subtypes)
choice.i	optional	Yes	display - can contain variables	Contains the ith choice in a multiple choice or multiple selection question. The values of **i** in a question must start at 1 and be consecutive.
comment or comment.i	optional	Yes	display - can contain variables	Contains feedback provided to students upon grading. The **comment.i** field provides response-specific comments in multiple choice questions. The values of **i** in a question must start at 1 and be consecutive.

qu.x.y. <field name>	Required or optional	Can include HTML or MathML?	Field Type	Description / Function
hint.i	optional	Yes	display - can contain variables	Contains a hint. The values of **i** in a question must start at 1 and be consecutive.
solution	optional	Yes	display - can contain variables	Contains a worked solution. Shown only in Study Session assignments.
info	optional - user specified	No	internal directive	Contains descriptive information about the question. Multiple user specified fields and corresponding values can be contained in this statement.

Definitions

- **Internal Directive** fields are *commands* that are evaluated and performed by the system. They cannot contain HTML or MathML tags and must conform to the system syntax for algorithms, formulas, math expressions, or command switches.
- **Display** fields hold question information that is used in displaying the question or related feedback, hints, or solutions. The **topic** and **name** fields cannot contain variable values because algorithmic variables are set at the level of the question.

Question Modes (Question Types)

Every question must begin with a **mode** statement. Using the **mode** statement, Maple T.A. formats the question on the page.

- Different question types require different fields in their definition. For example, you must specify every choice for a multiple choice question as a choice.i field, whereas for a formula question you specify only the correct numeric answer.
- The question type is entered in the **mode** field. The question type specification can be all lower-case letters or have initial upper-case letters. For example, the following line would designate the fifth question in topic 3 as multiple choice:

```
qu.3.5.mode=multiple choice@
```

Table 8.6 lists all the Maple T.A. question types and their corresponding **mode** strings.

Table 8.6: List of Question Types and Corresponding modes

Question Type	mode string qu.x.y.mode=<question_type>@
Blanks [text] [menu] [formula] options	`qu.x.y.mode=Blanks@`
Clickable imagemap	`qu.x.y.mode=Clickable Image@`
Constant of integration/Formula mod C	`qu.x.y.mode=Formula Mod C@`
Customized symbolic palette	`qu.x.y.mode=palette@`
Equation	`qu.x.y.mode=Equation@`

Question Type	mode string qu.x.y.mode=<question_type>@
Essay	qu.x.y.mode=Essay@
Formula	qu.x.y.mode=Formula@
Formula with physical units	qu.x.y.mode=Dimensioned formula@
Key words	qu.x.y.mode=Key Words@
List	qu.x.y.mode=List@
Maple-graded	qu.x.y.mode=Maple@
Matching	qu.x.y.mode=Matching@
Matrices with numeric/formula entries	qu.x.y.mode=Matrix@
Multipart	qu.x.y.mode=Multipart@
Multipart formula	qu.x.y.mode=Multipart Formula@
Multiple choice	qu.x.y.mode=Multiple Choice@
Multiple selection	qu.x.y.mode=Multiple selection@
Non-permuting multiple choice	qu.x.y.mode=Non Permuting Multiple Choice@
Non-permuting multiple selection	qu.x.y.mode=Non Permuting Multiple Selection@
Numeric (with or without units)	qu.x.y.mode=Numeric@
Ordered list of formulas	qu.x.y.mode=Formula List@
Question Designer	qu.x.y.mode=Inline@
Restricted formula	qu.x.y.mode=Restricted Formula@
Short phrase	qu.x.y.mode=Short phrase@
True-or-false	qu.x.y.mode=True False@
Unordered list of formulas	qu.x.y.mode=Multi Formula@
Vector	qu.x.y.mode=Ntuple@

For more information on each question type, see *Question Types (page 131)*.

Required Question Elements

Each question must contain the following elements:

- Question mode
- Question statement
- Answer

In the following descriptions, the variables **x** and **y** represent the topic and question numbers.

Question Mode

Format: qu.x.y.mode=<question_type>@

The required **mode** statement declares the type of question. You must include it as part of the opening statement for the question.

Question Statement

Format: qu.x.y.question=<question_text>@

The required **question** statement holds the text of the question. Question statements can include any combination of HTML tags for formatting text, integrated HTML graphic or web media references, or MathML expressions.

Answer

Format: qu.x.y.answer=<correct_answer>@

The required **answer** statement specifies the correct answer value for the question. For more information on this field, see *The answer Field (page 127)*.

Example

```
qu.1.topic=Basic Topic 1@

qu.1.1.mode=Multiple Choice@
qu.1.1.question=What is the sum of 5 and 7?@
qu.1.1.answer=1@
qu.1.1.choice.1=12@
qu.1.1.choice.2=35@
qu.1.1.choice.3=2@
qu.1.1.choice.4=11@
qu.1.1.choice.5=-2@
```

See Also:

Question File Structure Details (page 119)

The name Field: Using Question Descriptions

The **name** field allows you to provide a description for each question. A descriptive title will help you identify individual questions from lists of questions shown in the **Question Editor** and the **Assignment Editor**. The **name** field is not required, however it is recommended.

The **name** field accepts HTML formatting.

Example

```
qu.x.y.name=Travel time for block on inclined plane  #1@
```

Note: If you omit the **name** field, or leave the description blank, the system substitutes the question type for the description. This can result in questions that show many similar items (for example, Multiple Choice, Multiple Selection, Multiple Choice, Multiple Selection, and Multiple Selection).

The answer Field

All question types in the system (excluding Maple-graded, Matching, and Essay) must contain an **answer** field that indicates the criterion for grading the question. The **answer** field is a *command* that the system evaluates and performs, if necessary. The output is compared against student responses for grading.

- Because the value of the **answer** field is processed by the system and compared to student responses, the **answer** field **must not** contain HTML, extraneous text (not part of the required answer), or special formatting.
- Answer field values for mathematics questions are processed by the system math parser, and must be entered in proper system math syntax or MathML. Student responses are processed by the parser and must reduce to mathematically equivalent values to be graded correct.

Table 8.7: Answer Field Requirements by Question Type

Question Type	Answer Field
Non-Numeric mathematics question types	Use the **?** operator in the **answer** field value to specify a margin of error or degree of tolerance in a student response. For more information, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.
Numeric questions	The value of the answer field must contain **answer.num** and **answer.units** definitions. You can assign null to the **answer.units** field, that is, **answer.units=@**, for answers without units. For Numeric questions, use the specialized answer tolerance fields. See *Numeric (page 146)* . For general information on tolerance, see *Controlling Answer Tolerance (page 183)*.
Multiple Choice, Multiple Selection, True-or-False, and Clickable Image questions	The value of the answer field indicates which response(s) are correct.
Free-response questions with text fields (for example, Blanks - [text] option, Key Word, or Short Phrase questions)	The value of the **answer** field must be a text string that is used to evaluate the correctness of a student response. For more information, see *Rules for Grading Free Response Questions (page 158)*
Maple-graded questions do not contain an **answer** field	You must specify Maple code to evaluate the student response. For Maple Formula questions, the system parses the student response, and then passes the parsed result to Maple. For Maple Syntax questions with the text entry response option, the student response is passed unparsed to Maple. For more information, see .
Matching questions do not contain an **answer** field.	The matching items are defined using pairs of statements: **term.i=<label_i>@** **term.i.def.1=<correct_match_of_label_i>@** where the values of **i** in a question must start at **1** and be consecutive.
Essay questions do not contain an **answer** field.	They are not graded by the system.

If a student response is incorrect, the default system behavior is to display the **answer** field value. For more information on this behavior, see *Tutorial: Displaying Answers vs. Formulas for Answers (page 212)*. To display custom feedback, use the comment field.

See Also:

The Answer Field Evaluation and Requirements (page 239)

Basic Math Syntax in the System (page 180)

The info Field

The **info** field allows you to add information subfields to a question to indicate subtopic, learning objective, level of difficulty, authorship, ownership, or other information. The **info** field is not required.

The **info** field values:

- Allow sorting and filtering in the question repository and during assignment creation
- Facilitate performance reporting by information field category (for example, assignment results by learning objective)

Example

```
qu.1.1.info=
  subtopic=Subtraction;
  difficulty=Easy;
  learningobj=Subtraction of two small integers;
  instructor=John Smith;
  course=Math  100
@
```

Hints, Coaching, and Performance Feedback

In a **hint** field, you can enter a hint that is displayed in assignments. (For information on adding hints using the **Question Editor**, see *Adding and Editing Hints (page 83)*.)

As an instructor:

- You define hints in questions.
- You enable the display of hints in Step 3 - Rules & Policies of the **Assignment Editor** so that a student can view them during an assignment.
- You can enable the display of hints in **any** assignment mode. The system also displays hints after grading if a student response is incorrect.
- If you do not enable hints, they are displayed only as feedback when a question is graded incorrect.
- You can create any number of hints for a question.
- **Important:** When creating questions that have a specific number of significant digits in the correct answer, it is recommended that you use the sig function to create algorithmic variables. By using these variables in the question statement, feedback, hints, and solution, ensure that values are displayed with the correct number of significant digits.

If you provide hints on an assignment, student can view the hints for a question at any time during an assignment session prior to submitting questions for grading by clicking the links for the numbered hints, for example, **Hint 1** and **Hint 2**. The hints will appear as hyperlinks below the question answer region.

Example

```
qu.x.y.mode=formula@
qu.x.y.question=What is the derivative of x<SUP>2</SUP>+3x+1?@
qu.x.y.answer=2x+3@
qu.x.y.hint.1=Differentiate the expression term by term.@
qu.x.y.hint.2=The derivative of x<SUP>2</SUP>  is 2x.@
qu.x.y.hint.3=The derivative of the constant term is zero.@
```

See Also:

Configuring Feedback Options (page 43)

Comments and Performance Feedback

Providing performance feedback is a two step process in the Maple T.A. system.

Step 1: Create content using the **comment** field for each question. (For information on adding a comment using the **Question Editor**, see *Adding and Editing Comments (page 78)*.) This topic discusses this necessary first step.

Important: When creating questions that have a specific number of significant digits in the correct answer, it is recommended that you use the sig function to create algorithmic variables. By using these variables in the question statement, feedback, hints, and solution, you ensure that values are displayed with the correct number of significant digits. For more information on the sig function, see *decimal(n, x), sig(n, x), int(x) (page 223)*.

Step 2: When creating an assignment, specify if and how feedback is displayed in Step 3 - Set Policies of the **Assignment Editor**. You can display information including the:

- Final grade for the test
- Correct answer for each question
- **comment** for each question

You can also exclude question-by-question grading and question-specific feedback from the final assignment reports displayed to a student.

Step 1 Details

The comment Field

To return custom feedback when a student enters an incorrect response, each question type supports a **comment** field. The comment is displayed *after grading*. In the **comment** field, you can include additional explanations about solution method or final answer.

You can create response-specific comments for multiple choice questions, in which a distinct comment is displayed for each choice. For more information, see *Multiple Choice Questions (page 163)*.

If you do not specify a comment for a question, the system displays *only* the *correct answer*.

For details on using the feedback in assignments, see *Assignment Editor Step 3: Setting Policies (page 30)*.

Using Complete Solutions (Study Session Assignments)

Note: Information in the **solution** field is available only in **Study Session** assignments. To display feedback, for example, step-by-step solutions, in all assignment modes, use the **comment** field.

In the **solution** field, you can enter a worked solution that is displayed in **Study Session** assignments. (For information on adding a complete solution using the **Question Editor**, see *Adding and Editing Comments (page 78)*.) In these sessions, the student can view the solution *before* entering an answer or after *submitting* the question to be graded.

When the system grades a question in a **Study Session** assignment, it states whether the response was correct or incorrect. It does not display the correct answer. The solution can be an essential teaching component in the session.

Important: When creating questions that have a specific number of significant digits in the correct answer, it is recommended that you use the sig function to create algorithmic variables. By using these variables in the question statement, feedback, hints, and solution, you ensure that values are displayed with the correct number of significant digits.

Example

```
qu.x.y.mode=formula@
qu.x.y.question=
What is the derivative of x<SUP>2</SUP>+3x+1?@
qu.x.y.answer=2x+3@
```

```
qu.x.y.hint.1=Differentiate the expression term by term.@
qu.x.y.hint.2=The derivative of x<SUP>2</SUP>  is 2x.@
qu.x.y.hint.3=The derivative of the constant term is zero.@
qu.x.y.solution=
d/dx( x<SUP>2</SUP> + 3x + 1  ) =
(d/dx)x<SUP>2</SUP> + (d/dx)3x  + (d/dx)1 =
2x + 3
@
```

Special Characters in Questions

The system supports the use of text, HTML, and MathML for the display of special characters and formatting in the following question fields.

- `topic`
- `name`
- `question`
- `choice`
- `hint`
- `solution`
- `comment`

For maximum compatibility, use only character tags from the HTML 4.0 specification.

MathML tags are recommended for special scientific characters such as:

- Algebraic symbols
- Greek letters (and other variable labels)
- Equation elements, including operators and arrows

The system uses a cross-platform MathML rendering technology to ensure that special characters rendered in MathML appear consistently on W3C®-compliant browsers.

To generate MathML tags that encode a specific character, use one of the following authoring methods.

- *LaTeX Authoring Overview (page 247)*
- *MathML Authoring with MathType (page 197)*
- *Editing with the Equation Editor (page 188)*

For other special characters (for example, non-English characters) you can use any acceptable HTML tabs, employing the Unicode expression for the character. Note that some platforms and browsers that are non-standard do not map Unicode HTML expressions correctly, so results can be unpredictable. Therefore, it is recommended that you use MathML tags for all math and science content. For more information, see http://www.unicode.org.

Important: In other question fields (not listed here), you must *avoid* HTML and MathML tags, and limit field content to plain text only. Specifically, HTML and MathML tags in the **answer** and **algorithm** field values are *rejected* by the system. Typically these fields are used to provide commands to the system for processing the question or grading. Hence, they must be formatted according to the system command syntax.

9 Question Types

9.1 Question Types Available in the System

There are many different question types available in the system. The following list describes some of the most-used question types.

- Blanks — A blanks question is a type of fill-in-the-blanks question. It presents the student with a question with one or more blanks. The response areas can be either free response fields or selection from a drop-down boxes of possible answers.
- Clickable imagemap — A clickable image question presents an image with a number of hot spots. Students are required to identify the correct image element by clicking the appropriate hot spot.
- Formula — A formula question allows instructors to compare a student response (generally a mathematical expression) to a specified answer. It has several variants.
    - Formula mod C — accepts answer within an additive constant
    - Formula with physical units — requires units as well as a numeric value
    - Multi Formula or Formula List — accepts answers as a list of values (unordered or ordered)
    - Restricted Formula — the correct answer only uses basic operations (no logs or trig)
    - Equation — answer is an equation
    - Chemistry — accepts chemical equations
- Key word — A key word or phrase question searches the student's response for specified key words or phrases, ignoring other text.
- List — A list question is another type of fill-in-the-blanks question. It offers a few more advanced features than blanks questions, including the ability to give partial credit to certain answers.
- Maple-graded — A Maple-graded question uses the Maple mathematical software to generate algorithmic variables in questions, generate plots, and evaluate student responses. The Maple-graded question type gives you access to the computational power of Maple. You have access to many different kinds of mathematical objects. This question type allows for questions with complicated answers, questions with different possible answers, and questions requiring a powerful answer-equivalence checker. You can even find common errors and assign partial credit.
- Matching — A matching question displays two lists. The student must match each element of the first list with an element of the second list.
- Multipart — A multipart question can contain a variety of related question materials, data, and even case scenarios for different types of assessment and learning. A multipart question is built out of a number of individual questions, which can be of any question type.
- Multiple choice — A multiple-choice question can have any number of choices, and has exactly one correct answer. You can add additional incorrect choices after the question is initially created. The choices can be presented in static or permuting order.
- Multiple selection — A multiple selection question is similar to multiple choice questions, but the question can have more than one correct answer. The choices can be presented in static or permuting order. Students must select all correct answers to get full credit. Otherwise, the score is calculated using a formula that takes into account the number of correct and incorrect answers.
- Numeric — A numeric question compares a student's response to a given number, with or without units. You can require the student response to a certain precision or within a certain margin of error.
- Palette — A palette question enables you to program a customized palette of symbols for the entry of student responses.
- Question Designer — A Question Designer question is created through the Question Editor. The Question Designer is a powerful and flexible question container that can hold multiple response areas. The response areas are answer

regions supporting various question types all in a single question. The Question Designer also provides a flexible layout environment, making it easy to include images, tables, and so on in your question.

- True or false — A true or false question is treated as a class of multiple choice question featuring only two choices.

You can include a Maple plot in any question type (including nonmathematical question types). For more information, see *Plotting Using Maple (page 201)*.

For a complete list of question types (nonmathematical and mathematical), see *Question Modes (Question Types) (page 124)*.

9.2 Math Question Types

Math Question Types - Overview

Basic Question Types

The fundamental math question type is **Formula**. This appears as a question statement, followed (on a separate line) by a single free-response entry cell that accepts a math formula, that is, an expression consisting of numbers, variable names, and the standard arithmetic operators and functions. The system grades it by comparing it with the correct answer. If the student response and the correct answer are algebraically or numerically equivalent, the response is graded correct. To require that the response have a specified form, use the **form** subtype. To create a formula question with multiple input cells to hold closely related responses to the same question statement, use the **multipart formula** question type.

The **Blanks [formula]** question type is closely related to **Formula**. It uses the same grading and entry options, but presents the response cell as an in-line entry cell for students to complete.

The **Numeric** question type accepts numeric quantities or numeric quantities with units in student responses, and automatically processes scientific notation expressions.

The **Dimensioned Formula** question type accepts a combination of numeric quantities and physical units. It recognizes equivalent physical units. For example, if 500g is specified as the correct answer, 0.5kg is also graded correct.

- Any math question type can include algorithmically rendered plots. The **Maple-graded** question type vastly expands the math and scientific visualization capabilities of the system. You can include a Maple plot in any question type (including nonmathematical question types). (For more information, see *Plotting Using Maple (page 201)*.) In Maple questions, you can specify Maple code to generate a plot based on the student response. (For more information, see *Plotting a Student Response (page 187)*.)

Other Math Question Types

- You can limit the range of standard math functions available. For example, the question "**What is sin(Pi/4)?**" should accept the answers **sqrt(2)/2** and **1/sqrt(2)** but not **sin(Pi/4)**. The **Restricted Formula** type limits the functions that the students can use to those commonly found on a non-scientific calculator (for example, the arithmetic operators and **sqrt**).

- The basic **Formula** type does not accept equations, that is, expressions consisting of two components separated by an "**=**" sign. For these questions (for example, to accept **y-3=4(x+1)** and **y=4x+7**), use the **Equation** type.

- Questions that have answers with an additive constant (for example, a constant of integration in indefinite integration questions) have many valid answers that differ by the addition of a constant quantity. The **Formula Mod C** type accepts all such answers as equivalent.

- To create a question that requires a response with multiple values or formulas in a specific order, use the Formula List question type.

- The **Multi Formula** question type accepts an unordered list of numbers or formulas separated by semicolons. A question that has a single formula answer is graded correctly if this type is used. It is recommended that you

use the **Multi Formula** type for questions that have a single correct answer, but, in general, have multiple correct answers because the **Formula** type presentation indicates to students that there is only one solution. For example, the only root of **x^2-2x+1** is **x=1** (with multiplicity two), but, in general, a degree two polynomial has two roots.

- To create a question that accepts responses that include a matrix, use the Maple Syntax question type. If you prefer to present a rectangular array of entry boxes for responses, use the **blanks[formula]** or Matrix type.
- For questions involving vectors and Cartesian coordinates, use the **Ntuple** type.

For a complete list of question types (nonmathematical and mathematical), see *Question Modes (Question Types) (page 124)*.

Math Question Types Comparison Table

This is a list of supported math question types.

Table 9.1: Math Question Types Comparison

Question Type	Description	Example Responses
Formula	Accepts numbers and formulas	$e^x \sin(x^2)$
Restricted Formula	Accepts numbers and formulas that do not include trigonometric functions or logarithms.	1/sqrt(2)
Equation	Grades equivalent equations as correct. (Other question types do not allow the "=" sign). **Note:** The expression given as the correct answer must consist of a single variable on the left-hand side of the equation and a formula that does not involve that variable on the right hand side. (That is, you cannot specify an implicit formula.) This restriction does not apply to students entering their answers.	y-2 = 4(x+1)
Blanks [formula] option	One or more entry boxes that accept numbers and formulas, inserted into a section of text. Blanks can be inserted into any standard HTML, including tables.	in-line formula expressions
Numeric	Accepts numbers in decimal form. Scientific notation is allowed but formulas are not allowed. Also allows you to specify required physical units.	$4.7 * 10^5$
Maple	For information, see *Maple-graded Question Type - Overview (page 140)*.	
Formula Mod C	Numbers and formulas. Accepts two answers as equivalent if they differ by a constant number.	x ln(x) x ln(x)+ 1
Multi Formula	Accepts an unordered list of numbers or formulas as a list of numbers and formulas separated by *semicolons*. The answer must match the terms of the list in any order. Explanatory text tells students to enter any order of numbers or formulas separated by semicolons.	4; 5x; 3
Ntuple	For questions about Cartesian coordinates or vectors. Accepts an ordered list of number or formulas separated by commas. (for example, a Cartesian point or a vector). Parentheses are recommended but not required. Explanatory text tells students to enter a list of formulas separated by commas enclosed in parentheses.	(2, 4t, 7t+1)

Question Type	Description	Example Responses
Formula List	Accepts an ordered list of numbers or formulas. Parentheses accepted but not recommended. Grading is same as for **Ntuple**. Explanatory text tells students to enter an ordered list of formulas or numbers separated by *commas*.	2, 4, 8, 16
Multipart Formula	Multiple questions arranged in parts (a), (b), (c), ... Otherwise operates like a formula question.	
Matrix	A rectangular array of entry boxes that accept numbers and formulas.	3 $4x^2$ sin(x) 6
Dimensioned formula	Accepts algebraic formulas, unit dimensions, numeric responses, or any combination of these.	**sin(x)** **6** **m** **6m** **cos(x) m/s**

Blanks [formula]

Question Mode: blanks [formula] option

```
qu.x.y.mode=blanks@
blank.<blank_number>=<math_expression>@
grader.<blank_number>=formula@
```

Description

The **blanks** question type is versatile. It can provide three different types of student response cells: text fields, drop-down boxes, and mathematical formula cells. The **blanks [formula]** option provides a means for asking a fill-in-the-blank question in which the answer is a mathematical formula. The text inside a blank is removed and a formula-entry cell is presented to students.

- The only difference between a **blanks [formula]** question and a **formula** or **multipart formula** question is the presentation of the entry cell. For **blanks** questions, the entry cell appears in-line with the question statement (rather than on a separate line as in a **formula** question).
- The **blanks [formula]** question type accepts only numbers and math expressions. There is no support for physical units.

To create a blanks [formula] question, specify the:

- **formula** argument in the **grader** field of the question
- correct answer in math syntax (for processing by the system math parser) in the **blank** field. To identify multiple correct answers, separate each with a comma. Do not use a trailing comma. See the following example.

Example Script for a Blanks Question that Uses All Three Blanks options:

```
qu.1.1.mode=Blanks@
qu.1.1.question=
  Select  "red" from the menu: <1>
  <br>
  Enter  "red" or "green" in the box: <2>
  <br>
  Enter  3 or 4 in the formula box: <3>
@
qu.1.1.blank.1=red,green,blue@
qu.1.1.blank.2=red,green@
```

```
qu.1.1.blank.3=3,4@
qu.1.1.grader.1=menu@
qu.1.1.grader.2=text@
qu.1.1.grader.3=formula@
```

Notes

- You must provide the correct answer in the system math syntax, **not** TeX. Enter the answer as you expect the student to enter it (although equivalent answers are graded correct).
- This option is similar to **Formula**, but may be more convenient for some questions.
- Another advantage of the **blanks** question type is that several blanks can be included in a single question, and different blank options can be mixed in a single question. This makes the **blanks** type powerful and flexible.
- You can use **blanks [formula]** questions to create a matrix question by setting the blank response cells in an HTML table.
- If the answer is a number (that is, does not have any variables), it is recommended that you specify in the question text whether an exact answer or an approximation is required. In the case of an approximation, the margin of error must be specified in your correct answer. This can be done by specifying a margin of error using the **?** operator.

  For example, if the answer field were

  ```
  {2.34 ? 0.01}
  ```

  any response between **2.33** and **2.35** (inclusive) would be graded correct. In other words, the number following the **?** operator is the margin of error. For more information on tolerances, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

See Also:

Mathematical Fill in the Blanks - [formula] option (page 268)

Free Response and Fill-in-the-Blank Questions (page 156)

Formula (page 137)

LaTeX Authoring Overview (page 247)

Multipart Formula (page 146)

Using Script Files to Author Questions (page 283)

Chemistry

Question Mode: Formula

```
qu.x.y.mode=Formula@
qu.x.y.grading=chemistry@
```

In the **chemistry** question type, student responses require a formula that matches the correct answer.

Table 9.2: Rules and Keys for Entering Chemistry Expressions

Expression	Keys
Superscripts and Subscripts	Enter superscripts using the caret ^ character, and subscripts using the underscore _.
Arrows in Equations	Use the text ->, <-, <=> for arrows.

Expression	Keys
Other Operators	Use * for the center dot operator.
	Use the + sign in equations.
	Use the - sign to indicate ion charges
	No other operations are allowed in equations.
Physical States, Ion charges, and Parentheses	Be sure to include physical states (in parentheses) if your equation requires them. Use the + and - characters for polarity and ion charges Use parentheses to clarify interpretation of groups of characters

See Also:

Mathematical Formula Questions in the Question Editor (page 98)

Entering Chemistry Expressions (page 186)

Constants of Integration Questions

Question Mode: Formula Mod C

```
qu.x.y.mode=Formula Mod C@
```

In the **Formula Mod C** question type, student responses require a formula that matches the correct answer to within an additive constant (+**C**).

This type is designed for indefinite integration problems, in which the answers differ by a constant of integration. Do not include the constant in the correct answer.

In a test, a note telling the student they do not need to include the constant of integration appears after the input field, as shown in **Figure 9.1**.

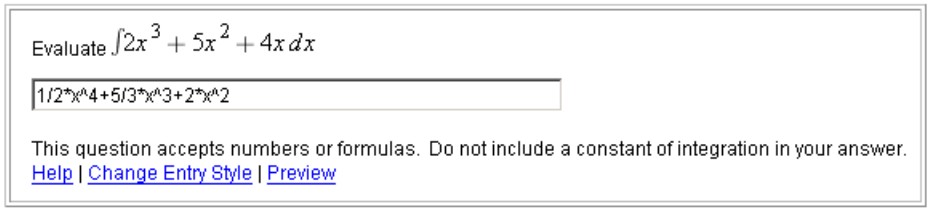

Evaluate $\int 2x^3 + 5x^2 + 4x\,dx$
1/2*x^4+5/3*x^3+2*x^2
This question accepts numbers or formulas. Do not include a constant of integration in your answer.
Help | Change Entry Style | Preview

Figure 9.1: Constants of Integration Question

The response 1/2*x^4+5/3*x^3+2*x^2+1 is also graded correct.

See Also:

Mathematical Formula Questions in the Question Editor (page 98)

Constants of Integration Questions in LaTeX (page 269)

Equation

Question Mode: equation

```
qu.x.y.mode=equation@
```

Use the **equation** question type for any question that requires an equation as an answer. It is the only question type that requires the inclusion of an "=" sign in the student response (and in the answer). You can use the full range of functions in the equation, that is, all trigonometric functions, **log**, **ln**, **abs**, and **sqrt**.

In a test, the input field for the response appears as shown in **Figure 9.2**.

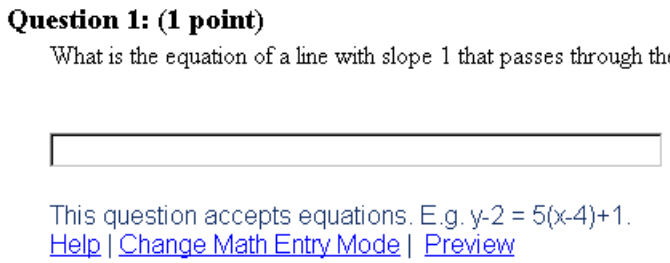

Question 1: (1 point)
What is the equation of a line with slope 1 that passes through the origin?
This question accepts equations. E.g. y-2 = 5(x-4)+1.
Help | Change Math Entry Mode | Preview

Figure 9.2: Equation Question

See Also:

Mathematical Formula Questions in the Question Editor (page 98)

Equation Questions in LaTeX (page 270)

Numeric (page 146)

Formula

Question Mode: formula

```
qu.x.y.mode=formula@
```

Use the **formula** type for any question where the student does not specify physical units in their response, and where the response is not an equation. You can use the full range of functions in the formula, that is, all trigonometric functions, **log**, **ln**, **abs**, and **sqrt**.

To create a formula question with multiple response cells in the same question, use the **multipart formula** question type. (**Note:** The **multipart formula** question type is different from the **multi formula** question type.)

In a test, the input field for the response appears as shown in **Figure 9.3**.

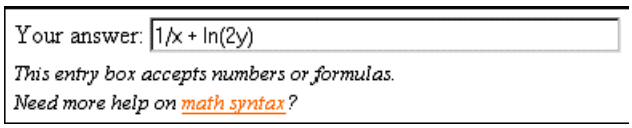


Figure 9.3: Input Field for a Formula Question

Note:

- If the answer is a number (that is, does not have any variables), it is recommended that you specify in the question text whether an exact answer or an approximation is required. In the case of an approximation, the

margin of error must be specified in your correct answer. This can be done by specifying a margin of error using the **?** operator.

For example, if the answer field were

```
{2.34 ? 0.01}
```

any response between **2.33** and **2.35** (inclusive) would be graded correct. In other words, the number following the **?** operator is the margin of error. For more information on tolerances, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

The following sub-types for this question type are available by specifying the **grading** field:

- Chemistry
- Form Variant (without simplification)

See Also:

Blanks [formula] (page 134)

Formula Questions in LaTeX (page 271)

Algorithmic Mathematical Formula Question Example Script (page 284)

Mathematical Formula Questions in the Question Editor (page 98)

Multipart Formula (page 146)

MultiFormula (page 145)

Formula with Units (page 139)

Numeric (page 146)

Formula - Form Variant

Question Mode: Formula (with "form" grading)

```
qu.x.y.mode=Formula@
qu.x.y.grading=form@
```

The **Formula - Form Variant** question type is a restricted-grading variant of the Formula question type. In this mode, the student response is required not only to be mathematically equal to the correct answer, but also to be expressed in the same form.

Thus

```
(x-1)(x+1),
(x+1)(x-1),
```

and

```
(x^2-1)
```

would be graded the same by an standard formula question. However, with the form submode set, only the first two would be graded as equivalent.

Authoring of this question variant is not directly available using the **Question Editor**. To invoke this grading variant, you must create a formula question, and then edit the source by adding a line: "grading=form@".

You can also specify it in LaTeX or the source for a question. When editing the source, include a line of the form "qu.1.1.grading=form@"

See Also

Formula (page 137)

Formula List

Question Mode: Formula list

```
qu.x.y.mode=Formula list@
```

The **Formula list** question type requires an ordered list of formulas in the student response. Use this for a list of formulas or numbers that is required in a *particular order*. You can use the full range of functions in the formulas, that is, all trigonometric functions, **log**, **ln**, **abs**, and **sqrt**.

Important: You must use **commas** to separate items in the correct answer (for example, **1,12,12**). If you use semicolons in the correct answer, the system treats the question as an unordered list (though the type is **Ordered**), and allows the student to give the formulas in any order.

In a test, the input field for the response appears as shown in **Figure 9.4**.

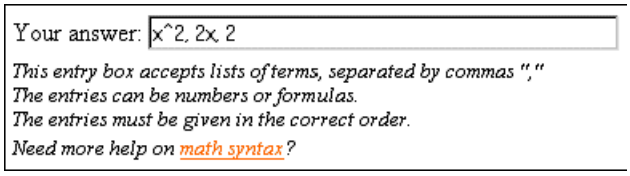

Your answer: x^2, 2x, 2
This entry box accepts lists of terms, separated by commas ","
The entries can be numbers or formulas.
The entries must be given in the correct order.
Need more help on math syntax?

Figure 9.4: Input Field for a Formula List Question

See Also:

Formula List Questions in LaTeX (page 271)

Formula (page 137)

Mathematical Formula Questions in the Question Editor (page 98)

MultiFormula (page 145)

Formula with Units

Question Mode: Dimensioned formula

```
qu.x.y.mode=Dimensioned formula@
```

The **Dimensioned formula** question type is a highly versatile. Correct answers can include algebraic formulas, unit dimensions, numeric responses, or any combination of these.

Use this question type for any formula where you require the student to specify the physical units for the expression (for example, **m/s** or **kg/cm^2**), and where the formula is not an equation. When you give the units with the answer, you must use the units that are recognized by the system.

The system can convert between units of the same type. For example, if you type the answer as **30s**, the student can give the answer as **30s**, **0.5min**, or **1/120 hour**. In each case, the system grades the response correct.

In a test, the input field for the answer appears with a link provided to a list of the recognized units.

See Also:

Formula (page 137)

Using a Table of Units (page 205)

Mathematical Formula Questions in the Question Editor (page 98)

Maple-graded Question Type - Overview

The Maple-graded question type uses the **Maple** computer algebra system to generate algorithmic variables in questions, generate plots and evaluate student responses. The Maple-graded question type gives you access to the computational power of Maple. It includes facilities for algebra, calculus, differential equations, discrete mathematics, graphics, numerical computation, and many other areas of mathematics.

Since the Maple engine is used when grading student's responses, Maple T.A. will give credit for any response that is mathematically equivalent to the correct answer.

Using the Maple-graded question type, you have access to many different kinds of mathematical objects, not just simple expressions. You can use Maple to create questions whose responses require sets, differential equations, unevaluated integrals, groups and many other types of mathematical data. The Maple-graded question type allows for questions with complicated answers, questions with different possible answers, and questions requiring a powerful answer-equivalence checker.

With the Maple-graded question type, you also have access to the plotting capabilities of Maple. You can use Maple to plot a student response (or a function derived from a student response, for example, the definite integral of the student response) for a Maple-graded question type or display a plot for any question type.

You can assign partial grades, allowing you to find common errors and reward partial credit.

You can also use functions and routines that are contained in a separate Maple Repository (Maple Library). You can access your personal Maple library archives from within your Maple questions.

Maple-graded Formula & Maple Syntax Subtypes

The Maple-graded question type has two subtypes:

- Maple-graded Formula
- Maple Syntax

Both the Maple-graded Formula and Maple Syntax subtypes allow you to use Maple functions and expressions, for example, trigonometric functions, **log10, ln, abs, sqrt, diff, int**, and **LinearAlgebra[Determinant]**. In addition to the full range of Maple functions, you can use programming constructs to evaluate responses.

Difference Between Maple Formula and Formula Question Types

Formula questions are very useful when the correct response is a straightforward mathematical expression using standard common functions, and when the correct answer is easy to describe and easy to evaluate.

The Maple-graded Formula subtype differs from the Formula question type.

- When writing a Maple-graded Formula question, you can use Maple functions and expressions in the evaluation of the student response.
- When writing a Formula question type, you *cannot* use Maple code in the evaluation of the student response.

Viewing your source code reveals that for Maple-graded questions (both Maple Formula and Maple Syntax subtypes) **mode=Maple@**, while for Formula question types **mode=formula@.**

Note: There is no difference for the student in syntax or range of expressions.

Difference Between Maple-graded Formula and Maple Syntax with Text Entry Mode in Student Response

These two subtypes are distinguished by a student's response.

- A student should *not* use Maple commands and expressions in a response to a question generated using Maple-graded Formula. For details, see *Maple-graded Formula (page 142)*. The student responds using other Maple T.A. formula question syntax.
- A student is *required* to use Maple commands and expressions in a response to a question generated using Maple Syntax with the Text entry response option. For details, see *Maple Syntax (page 143)*.

Difference Between Maple-graded Formula and Maple Syntax with Text Entry Mode in Syntax Checking

These two subtypes are distinguished by syntax checking of the student response.

- The system verifies the syntax of a student response for a Maple-graded Formula question using a basic syntax checker. For details, see *Maple-graded Formula (page 142)*.
- The system does not verify the syntax of a student response for a Maple Syntax question. As the author, you can verify whether the student is using a Maple command to calculate the answer. For details, see *Maple Syntax (page 143)*.

Difference Between Text Entry and Symbolic Entry for Student Response in Maple Syntax Questions

With a Maple Syntax question, you can choose whether the students enter responses using Text mode or Symbol mode. In both cases, the response is sent to Maple, where you can apply sophisticated grading methods to evaluate the student response.

- A student can enter math expressions, including Matrices, in a natural way using Symbol mode. For example, a student can enter a Matrix using a palette.
- A student is *required* to use Maple commands and expressions in a response to a question generated using Maple Syntax with the Text entry response option.

For details, see *Maple Syntax (page 143)*.

Guidelines for Using Maple Code for Maple-graded Questions

- A Maple-graded question must use valid Maple code to evaluate the answer. Complete each line of code with a semicolon.
- The last line of your question code must evaluate to a Boolean value (**true** or **false**) or a floating-point number between 0.0 and 1.1 for partial grading. In many cases, it is recommended that you use the Maple **evalb** command.
- Use the long form name for all package functions, for example, **VectorCalculus[ArcLength]**.

For an example that uses partial grading, see *Applying Partial Grading to Maple-graded Questions (page 97)*.

Providing Feedback to Students

- To display the correct answer as feedback for a student response, you must enter the correct answer in the **Enter Maple code that evaluates to the correct answer** field.
- For open-ended questions, such as "Give and example of an increasing function on the interval [0, 10]", it is recommended that you include a comment for a graded response. Otherwise, no comment is displayed as feedback to the student if the response is incorrect. Comments are entered in the Question Name & Type screen.

Maple-graded Question Type Help Pages

The Algorithm Designer (page 80)

Maple-graded Questions in the Question Editor (page 96)

Displaying a Maple Plot (page 201)

Maple Syntax Question Example Script (page 294)

Math Question Types Comparison Table (page 133)

Mathematical Functions & Operations (page 204)

Plotting a Student Response (page 187)

Tutorial: Maple-based Algorithmic Question Authoring (page 244)

Using a Maple Repository (page 90)

Using Maple Code to Prevent Cheating in Maple Syntax Questions (page 185)

Using the Comment Field in a Maple Question (page 294)

Maple-graded Formula

Question Mode: Maple

```
qu.x.y.mode=Maple@
type=formula@
```

In Maple-graded Formula question types, students can enter a math formula, that is, an expression constructed of numbers, variable names, and the standard arithmetic expressions and functions.

Note: In the Question Editor, on the Maple-graded question screen, the **Select the type of expression you want to accept** field lists the Maple-graded Formula option as simply **Formula**.

Guidelines

- The student *should not* use Maple commands in the response. For an explanation, see **Syntax** and **Student Responses** below.
- The instructor must write code such that the student does not have to use a trailing semicolon in the response.

For example, **evalb($RESPONSE=factor(x^2-1));** as a single line of code allows the student to respond with **(x+1)(x-1)** omitting the trailing semicolon. This is especially important, as questions created using the Maple-graded Formula question subtype are not obvious Maple questions from the student's perspective. A non-Maple student user can answer a Maple-graded Formula question type without knowing Maple syntax.

Syntax

Maple T.A. verifies the student response using a basic syntax checker.

- If the answer is entered in Text mode, the formula syntax checker assesses the response, changes expressions to Maple syntax, for example, 2x becomes 2*x, and reports errors if the expression is not a formula. It then generates a Maple statement.
- If the answer is created using the Equation Editor, a MathML expression is parsed, the syntax checker is run to ensure the answer is of type formula, and then a Maple statement is generated.

Student Responses

You do not need to explicitly check that the student response is a formula. For example, you do not need to check whether **factor(expr)**is a command that calculates the expression for the student. However, a student response like **factor(x^2-1)** would be interpreted as implicit multiplication and converted to **f*a*c*t*o*r*(x^2-1)**. As such, the student response will be marked as *wrong*.

You may need to write Maple code for responses that are interpreted as formulas by Maple T.A., but are Maple commands. For example, if you ask for the value of **sin(Pi/4)**, you must use code to ensure that the student does not simply enter **sin(Pi/4)**, which the Maple T.A. syntax checker accepts and passes to Maple. The correct response is, for example, **sqrt(2)/2**, which is also a Maple T.A. formula and a Maple command.

Variables

The variable **e** should *not* be assigned as a global variable in **Maple** code. It can be used, however, as a local variable.

See Also:

Maple-graded Questions in the Question Editor (page 96)

Maple-graded Question Type - Overview (page 140)

Maple Syntax

Question Mode: Maple

```
qu.x.y.mode=Maple@
type=maple@
```

Maple Syntax is a subtype of the Maple-graded question type.

In this subtype, the student is required to :

- Respond using Maple commands and expressions
- Enter syntactically correct Maple expressions

In a Maple Syntax question, you must specify whether the response area is a text-entry area or a symbolic-entry area.

Example Uses

The Maple Syntax subtype is particularly useful when the answer is to be expressed as a:

- Definite integral, **int(f(x),x=a..b)**
- Derivative, **diff(f(x),x)**
- Limit, **limit(f(x), x=0)**
- Differential equation, for example, **diff(y(x,t),t,t)=c^2*diff(y(x,t),x,x)**
- Set, for example, **{1,4,9}**
- Matrix, for example, $\begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$

Some example questions involving matrices are created in *Tutorial: Maple Syntax Questions with Matrices (page 107)*.

Maple Syntax with Text Entry Response Option

```
qu.x.y.mode=Maple@
type=maple@
allow2d=0
```

Guidelines

Student Use of Unevaluated Forms of Maple Commands

Some Maple commands take unevaluated forms. For example, **Diff** is the inert form of **diff**. In Maple T.A., students are directed to answer questions using the unevaluated forms of Maple commands. Your code should accept these forms.

Semicolon Usage

The instructor should specify to the student (in the question text) whether a trailing semicolon is required, or alternatively, write code such that the presence or absence of a trailing semicolon has no bearing on the evaluation of the response.

For example, **evalb($RESPONSE=factor(x^2-1);** as one line of question code does not allow a student response with a semicolon, such as **(x-1)*(x+1);** The response is marked as *incorrect*.

If the question code contains two lines, completed with semicolons, such as

```
A:= $RESPONSE;
evalb(A=factor(x^2-1);
```

the student can answer with or without a semicolon and the response is marked as *correct*.

Syntax

Maple T.A. does *not* check the syntax of the student response.

- If the answer is entered in Text mode, it is sent directly to Maple for parsing. The student is expected to enter syntactically correct Maple expressions. Shortcuts such as 2x are marked incorrect.

As the author, you can verify whether the student is using a Maple command to calculate the answer. For more information, see *Using Maple Code to Prevent Cheating in Maple Syntax Questions (page 185)*.

Maple Syntax with Symbolic Entry Response Option

```
qu.x.y.mode=Maple@
type=maple@
allow2d=2
```

For a tutorial on creating a Maple Syntax question with Symbolic mode response, see *Tutorial: Maple Syntax Questions with Matrices (page 107)*.

Syntax

Maple T.A. parses the student response entered in Symbol mode to convert the entered expression into Maple syntax, which is sent to Maple.

Note that, since a translation of the symbolic entry into a Maple Input expression takes place, the technique of using "$RESPONSE" to check whether students entered a specific command may not be appropriate. This technique is only recommended for Maple Syntax questions with the Text entry response option.

Variables

The variable **e** should *not* be assigned as a global variable in **Maple** code. It can be used, however, as a local variable.

See Also:

Maple-graded Questions in the Question Editor (page 96)

Matrix

Question Mode: Matrix

```
qu.mode.x.y=matrix@
```

The **Matrix** question type generates an **n** by **m** matrix of text fields, each of which accepts a number or formula. It is similar to the blanks[formula] question type.

Matrix questions must be created in LaTeX or in a plain text .qu file. You must specify a unique matrix as the correct solution.

Note: It is recommended that you use the Maple Syntax question type for questions that will have a matrix response. Maple Syntax questions provide more flexibility than the Matrix questions in question authoring and grading. With the Maple Syntax question type:

- Rather than showing the student with a specific size of matrix with a blank for each entry, the student can select the appropriate size of the matrix in their answer (for instance, using a Palette).
- By sending the entire matrix to Maple for grading, you can evaluate properties of the matrix (such as whether it is invertible or symmetric).
- Partial grading is possible.

See Also:

Matrix Questions in LaTeX (page 274)

Multi Formula

Question Mode: multi formula

```
qu.x.y.mode=multi formula@
```

In the **multi formula** question type, the correct student response is an *unordered* list of formulas. Use this for any unordered list of formulas or numbers. You can use the full range of functions in the formulas, that is, all trigonometric functions, **log**, **ln**, **abs**, and **sqrt**.

Important: You must use **semicolons** to separate the items in the correct answer (for example, **1; 12; 12**). If you use commas in the correct answer, the system treats the question as an ordered list (though the type is **Unordered**), and requires that the student give their response in the order in which you entered the correct answer.

In a test, the input field for the response appears as shown in **Figure 9.5**.

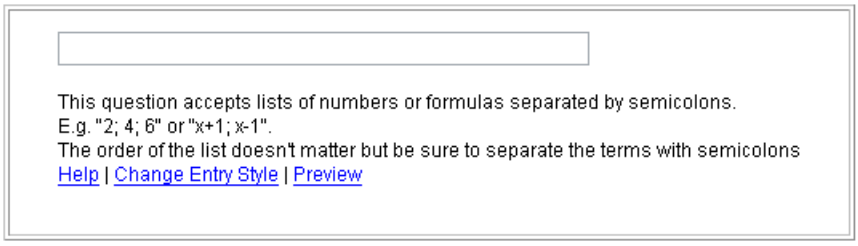

This question accepts lists of numbers or formulas separated by semicolons.
E.g. "2; 4; 6" or "x+1; x-1".
The order of the list doesn't matter but be sure to separate the terms with semicolons
Help | Change Entry Style | Preview

Figure 9.5: Input Field for a Multi Formula Question

See Also:

Formula (page 137)

Formula List (page 139)

Mathematical Formula Questions in the Question Editor (page 98)

Multi Formula Questions in LaTeX (page 274)

Multipart Formula

Question Mode: multipart formula

```
qu.x.y.mode=multipart formula@
```

The **multipart formula** question type behaves as a multipart question in which each part is a formula question. It displays multiple text fields for multiple student responses.

Note: A **multipart formula** question displays multiple individual text fields, while a **multi formula** question requires a student to enter multiple formulas in one text field as its correct answer.

Multipart formula questions must be authored in LaTeX or script files.

To identify multiple correct answers, separate each with a semicolon in your answer field definition statement. (Include a trailing semicolon for the last answer in the list.)

In a test, the input field for the response appears as shown in **Figure 9.6**.

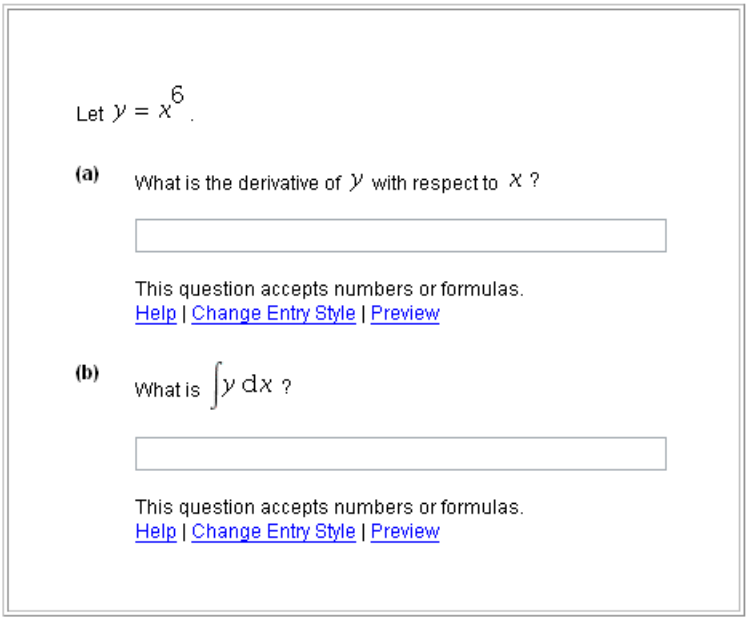

Let $y = x^6$.
(a)
What is the derivative of y with respect to x ?
This question accepts numbers or formulas.
Help | Change Entry Style | Preview
(b)
What is $\int y\,dx$?
This question accepts numbers or formulas.
Help | Change Entry Style | Preview

Figure 9.6: Input Field for a Multipart Formula Question

See Also:

Formula (page 137)

MultiFormula (page 145)

Numeric

Question Mode: Numeric

```
qu.x.y.mode=numeric@
```

The numeric question type accepts:

- Pure numbers (without physical units) in decimal form or scientific notation
- Numbers with units

Details:

- You cannot use equations or algebraic formulas in the correct answer.
- Equivalent numeric expressions and units are recognized. The system recognizes default unit equivalents. You can also create a question bank-specific table of equivalent units.
- Students are prompted to enter responses in scientific or numeric format, with or without physical units.
- You specify the inclusion of units in the correct answer using the **answer.units** question field. If you do not assign a value to the **answer.units** statement, the correct answer is dimensionless.
- Using the optional **showUnits** field, you control whether the system displays the question with the **Units** text field.
- You can control the precision and tolerance (or margin of error) required in student responses. The easiest way to set these modes is by using the Question Editor to author numeric questions.
- You can specify whether arithmetic is allowed in student responses.

Important: When creating questions that have a specific number of significant digits in the correct answer, it is recommended that you use the sig function to create algorithmic variables. By using these variables in the question statement, feedback, hints, and solution, you ensure that values are displayed with the correct number of significant digits.

- Single or multiple numeric questions can be included along with other response objects within a Question Designer question.

Example

```
qu.1.1.mode=numeric@
qu.1.1.question=
  <question_text>
@
qu.1.1.answer.num=<numeric part of the answer, for example, 3>@
qu.1.1.answer.units=<units part of the answer, for example, m/s>@
qu.1.1.showUnits=<true or false; whether to display the question with a Units text field>@
qu.1.1.grading=<grading mode; see below>@
qu.1.1.units=<(optional) reference to external table of units; see below>@
qu.1.1.digit=<# sig digits (positive integer);  required for exact_sigd and toler_sigd grading
 modes>@
qu.1.1.err=<error/tolerance; required for toler_sigd and toler_abs grading modes>@
qu.1.1.perc=<percentage error/tolerance; required for toler_perc grading mode>@
```

Grading Modes

Using the grading mode, you can control the precision or margin of error in the student responses graded correct.

Precision

exact_sigd

The numeric value of the student response must match **answer.num**.

The number of significant digits in the student response must match **digit**.

For example, if **answer.num = 2.5** and the **digit = 3**, only **2.50** is correct. In particular, **2.5** is incorrect.

exact_value

The numeric value of the student response must match **answer.num**.

For example, if **answer.num = 2.5**, **2.50** and **2.5** are correct, but **2.500001** is incorrect.

Margin of Error

`toler_perc`

The numeric value of the student response must be within **perc %** of **answer.num**.

For example, if **perc = 5** and **answer.num = 2.5**, any response between **2.5 * 0.95** and **2.5 * 1.05** is correct. In particular, **2.38888888** is correct.

`toler_abs`

The numeric value of the student response must be within **+/- err** of **answer.num**. The error can be specified as a numeric value or algorithmic variable.

For example, if **err = 0.25** and **answer.num = 2.5**, any response between **2.25** and **2.75** (inclusive) is correct. In particular, **2.333333333** is correct.

Precision and Margin of Error

`toler_sigd`

The numeric value of the student response must be within **+/- err** in the **digit**th significant digit of **answer.num** (inclusive) **and** have **digit** significant digits. The error can be specified as a numeric value or algorithmic variable.

For example, if **err = 2**, **digit = 3**, and **answer.num = 2.5**, the responses **2.48, 2.49, 2.50, 2.51,** and **2.52** are correct. All other responses, for example, **2.499** and **2.5**, are incorrect.

For example, if **err = 1**, **digit = 3**, and **answer.num = 27.18**, the responses **27.1** and **27.2** are correct. Although **27.3** is within **0.1** of **27.18** rounded to **3** significant digits, it is incorrect because it is not within **0.1** of **27.18**.

Using a Question Bank-specific Table of Units

To use a table of units, include the following statement:

```
qu.<topic_number>.<question_number>.units=<tablename>@
```

where **<tablename>** is a table of unit definitions in the question bank. For more information, see *Using a Table of Units (page 205)*.

See Also:

Numeric Questions in the Question Editor (page 102)

Numeric Questions in LaTeX (page 276)

Formula with Units (page 139)

Controlling Answer Format in Numeric Questions (page 182)

Controlling Answer Tolerance (page 183)

Questions with Plots

Basic System Plots

You can integrate basic function plotting or graphing components from algorithmic formulas in question statements or feedback. You must use LaTeX or plain text script authoring to create these questions.

For more information, see **Graph Plotting Applet**, below.

Maple Plots

Use the Maple computer algebra system to generate sophisticated two- and three-dimensional plots in your questions and feedback. See *Plotting Using Maple (page 201)*. You can add a Maple plot using any authoring method.

Graph Plotting Applet

The graph plotting applet is an interactive plot that is added to a question for dynamic plotting. The applet displays axes for the graph even when the axes lie outside of the viewable area.

- Click and drag the graph area to bring up a cursor that displays the coordinates of the current point
- **Shift**-click and drag the graph area to shift the viewable area
- **Ctrl**-click and drag the graph to re-scale the image

```
qu.1.topic=Plot@
qu.1.1.mode=Multi Formula@
qu.1.1.name=X-intercepts@
qu.1.1.editing=useHTML@
qu.1.1.question=<p class="noindent">What are the
          <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline">
            <mi class="MathClass-ord">x</mi></math>
          -intercepts of the graph shown?</p>
          <applet code="applets.grapher.Graph" width="350" height="350"  archive="graphing.jar">
            <param name="y1" value="(x-1)(x+3)(x-5)"></param>
            <param name="xMin" value="-6"></param>
            <param name="xMax" value="6"></param>
            <param name="yMin" value="-6"></param>
            <param name="yMax" value="6"></param>
            <param name="gridLines" value="12"></param>
          </applet>@
qu.1.1.answer=-3;1;5@
qu.1.2.mode=Multi Formula@
qu.1.2.name=X-intercepts@
qu.1.2.editing=useHTML@
qu.1.2.question=<p class="noindent">What are the
          <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline">
            <mi class="MathClass-ord">x</mi></math>
          -intercepts of the graph shown?</p>
          <applet code="applets.grapher.Graph" width="350" height="350"  archive="graphing.jar">
            <param name="y1" value="(x-1)(x-3)(x-5)"></param>
            <param name="xMin" value="2"></param>
            <param name="xMax" value="6"></param>
            <param name="yMin" value="-6"></param>
            <param name="yMax" value="6"></param>
            <param name="gridLines" value="12"></param>
          </applet>@
qu.1.2.answer=-3;1;5@
```

See Also:

Graph Plotting Applet in Questions (page 90)

Plotting a Student Response (page 187)

Authoring Questions with Embedded Plots Example Script (page 297)

Graph Plotting in LaTeX Questions (page 259)

Restricted Formula

Question Mode: Restricted formula

```
qu.x.y.mode=Restricted formula@
```

In the **Restricted formula** question type, the correct answer is a formula that uses only basic operations (for example, arithmetic operations and **sqrt**). You can use this for questions that require a formula without physical units that does not include trigonometric functions, **log**, **ln**, or **abs**, and which is not an equation.

When authoring in the system **Question Editor**, if you include a restricted function in the correct answer, the system warns you that the function is not allowed in **restricted mode**.

Note: If the answer is a number (that is, does not have any variables), it is recommended that you specify in the question text whether an exact answer or an approximation is required. In the case of an approximation, the margin of error must be specified in your correct answer. This can be done by specifying a margin of error using the **?** operator.

For example, if the answer field were

```
{2.34 ? 0.01}
```

any response between **2.33** and **2.35** (inclusive) would be graded correct. In other words, the number following the **?** operator is the margin of error. For more information on tolerances, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

See Also:

Mathematical Formula Questions in the Question Editor (page 98)

Equation (page 137)

Formula (page 137)

Restricted Formula Questions in LaTeX (page 279)

Vectors of Formulas

Question Mode: Ntuple

```
qu.x.y.mode=Ntuple@
```

The **Ntuple** question type requires a vector of formulas (a sequenced list) in correct responses. Use this for questions with vectors of formulas or numbers. You can use the full range of functions in the formulas, that is, all trigonometric functions, **log**, **ln**, **abs**, and **sqrt**.

- This is an ordered list. The entries must appear in the same order as in the given correct answer.
- The correct answer given should have parentheses, such as **(3,5)** or **(2,1,-3)**, because this is indicated in the instructions under the answer box. However, a response is graded correct with or without the parentheses.
- This type is graded in the same manner as the **Formula List** type. The only difference is the instructions given to the students.

In a test, the input field for the response appears as shown in **Figure 9.7**.

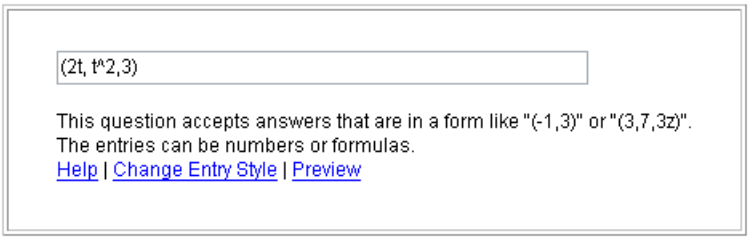

(2t, t^2,3)
This question accepts answers that are in a form like "(-1,3)" or "(3,7,3z)".
The entries can be numbers or formulas.
Help | Change Entry Style | Preview

Figure 9.7: Input Field for a Vector of Formulas Question

See Also:

Mathematical Formula Questions in the Question Editor (page 98)

Formula (page 137)

Formula List (page 139)

Coordinates of Points and Vectors Questions in LaTeX (page 280)

9.3 Applet Questions

Question Mode: Applet

```
qu.x.y.mode=Applet@
```

The **Applet** question presents a Java applet. Students are required to interact with the applet using the mouse and/or keyboard to alter the applet or identify hot spots in the movie to produce the correct answer.

```
qu.1.1.mode=Applet@
qu.1.1.question=The left-hand ball has mass of $m1 kg and the right-hand ball has mass $m2
 kg.  The center mass of the left ball is $h1 meters above the ground.  Move the right-hand
 ball to the height that gives it the same potential energy as the ball on the left.@
qu.1.1.answer=$h2@
qu.1.1.applet=applets.sliderBall.SliderBall@
qu.1.1.width=100@
qu.1.1.height=200@
qu.1.1.param.right=2@
qu.1.1.param.left=$h1@
qu.1.1.grader.class=gateway.question.grader.StringMatch@
qu.1.1.grader.param.ignoreCase=true@
qu.1.1.algorithm=$h1=int(rint(35)+5);
    $h2=int(rint(35)+5;
    $m1=rand(10,70,3);
    $m2=sig(3,$m1*$h1/$h2)@
qu.1.1.name.Physics Applet Question=2@
qu.1.1.editing=useHTML@
```

Applet Question Authoring

There are three steps to creating an Applet Question. Steps 1 and 2 require some familiarity with Java programming.

1. Create and deploy the Java applet.
2. Create and deploy a Java class grading.
3. Author the question.

Step 1: Create and Deploy a Java Applet

The applet is required to do three things:

4. Provide a **getResponse()** method which can be queried to find the current student response.
5. Call method **getParameter("response")** to read the **response** parameter for a string containing any previous attempt at a response to this question, and initialize the applet to display that state.
6. Call method **getParameter("locked")** to read the **locked** parameter, and disable user modifications if the flag is set.

You can also make additional **getParameter(**paramName**)** values to read any additional parameters for initialization. These are passed to the param fields in the question (see Step 3).

After creating the applet, place it in the **applets** directory or an appropriate subdirectory if packages are used) on the Maple T.A. server.

Step 2: Create and Deploy a Java Class for Grading

This is a server-side grader module that accepts the student response as prepared by the applet and compares it against the correct answer you will set in Step 3.

Your grader class should implement three methods:

7. An **init(String** paramName**, String** paramValue**)** method with void return type. This method receives a set of name/value pairs which it can use for initialization. These are passed from the grader.param fields in the question (see Step 3).
8. A **grade(String** correctAnswerText**, String** studentResponseText**)** method with a double return type value between 0.0 and 1.0, where 1.0 is full credit. The method should compare the text of the student response with text representing the correct answer.

   Note: The student response text is delivered from the applet's getResponse() method. In many cases, grading will be performed by judging whether the response is "the same" as the correct answer text. However, in some cases the correct answer text may only provide general guidelines for grading. For example, the correct answer might be "up and to the left" and the student response might be a point "(76,145)" that the student clicked on.
9. A **generateResponseMatchingAnswer(String** correctAnswerText**)** method with String return type. This method is called to get a sample correct answer to display as student feedback. The method must use the answer text to generate a string which the applet will display as the correct answer when passed the string as a response parameter.

   Note: In many cases, it will be enough to simply return the answer text in unmodified form. However, when the answer text only provides qualitative grading criteria ("up and to the left") it is more difficult to generate a sample response. In that case, it may be a good strategy to provide an "example" answer as a grader parameter in the testbank. For example, set the answer text to be "up and to the left" and set grader.param.example=(140,27).

After your grader class is created, create a JAR file and place it in the /WEB-INF/lib directory so it is accessible to the class loader. Then restart the Maple T.A. server.

Step 3: Author the Maple T.A. Question

The last step is to host your Java applet in a Maple T.A. question, using the Applet question type. Applet questions must be authored in a plain text script file.

For example, you can enter the sample code above for a question with slider balls in a plain text file, then save the file as **filename.qu**.

9.4 Blanks Questions

Blanks Questions can be authored inside the **Question Editor/Select Question Type** screen by selecting **Fill in the blanks** as the question type, or you can author them using LaTeX or plain-text script production paths.

To specify the form of the cell, use one of the following options:

- **[Menu] option.** Select a choice from a drop-down box. This variety of **Fill-in-the-Blank** questions can be authored to provide context-sensitive feedback. That is, you can provide a specific comment for each choice.

  If you author in a text or HTML editor, use the lines:
  qu.x.y.blank.<blank_num>=<correct_response>,<incorrect_1>,...,<incorrect_n>@
  qu.x.y.grader.<blank_num>=menu@
  in the question script. Use **<<blank_num>>**, for example, **<1>**, to mark the location of the drop-down box in the **question** statement.
- **[Text] option.** Free-response text fields If you author in a text or HTML editor, use the lines:
  qu.x.y.blank.<blank_num>=<correct_response_1>,...,<correct_response_n>@
  qu.x.y.grader.<blank_num>=text@
  in the question script. The correct responses are graded correct. Use **<<blank_num>>**, for example, **<2>**, to mark the location of the text field in the **question** statement.
- **[Formula] option.** Free-response math cells
  If you author in a text or HTML editor, use the lines:
  qu.x.y.blank.<blank_num>=<correct_formula_1>,...,<correct_formula_n>@
  qu.x.y.grader.<blank_num>=formula@
  in the question script. The correct formulas are graded correct. Use **<<blank_num>>**, for example, **<3>**, to mark the location of the formula cell in the **question** statement.

Examples of the three varieties of blanks questions are shown in **Figure 9.8**.

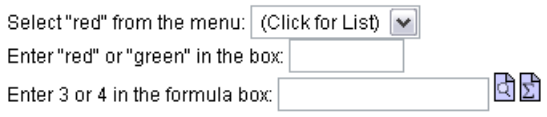


Figure 9.8: Three Varieties of Blanks Questions: Drop-Down List, Text Box, and Formula Box

9.5 Clickable Image Questions

Question Mode: clickable image

```
qu.x.y.mode=clickable image@
```

A **clickable image** question presents an image with a number of hot spots. Students are required to identify the correct image element by clicking hot spots. Using the **Question Editor** in the system, you can create these interactive questions by working with an image and identifying hot spot response regions by painting regions with the mouse.

A sample clickable image question is shown in **Figure 9.9**.

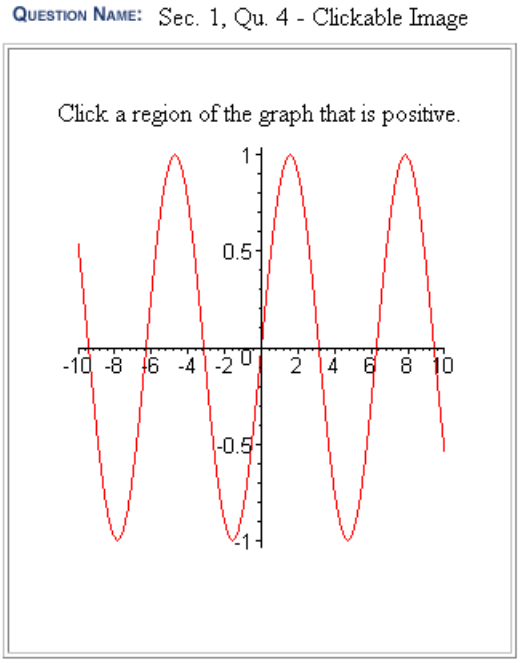


Figure 9.9: Clickable Image Question

See Also:

Clickable Image Question Example Script (page 288)

Images and Reference Files in Questions (page 87)

Clickable Image Questions in the Question Editor (page 93)

9.6 Ungraded Essay Questions

Question Mode: essay

```
qu.x.y.mode=essay@
```

An **essay** question is not automatically graded by the system. Student responses are sent to you, the instructor, who scores them traditionally, and assigns a grade in the system Gradebook. No answer or feedback is available to the student.

See Also:

Essay Question Example Script (page 289)

Essay Questions in the Question Editor (page 94)

Essay Question Example Script (page 289)

9.7 Flash Questions

Question Mode: Flash

```
qu.x.y.mode=Flash@
```

The **Flash** question presents a Flash movie of the question. Students are required to interact with the movie using the mouse and/or keyboard to alter the movie or identify hot spots in the movie to produce the correct answer.

```
qu.1.1.mode=Flash@
qu.1.1.movie=<path to your .swf file>@
qu.1.1.question=Move the red disk to the point ($x, $y).@
qu.1.1.answer=$x,$y@
qu.1.1.width=400@
qu.1.1.height=400@
qu.1.1.algorithm=
 $x = range(1,8);
 $y = range(1,8);@
qu.1.1.name=Graph Coordinates@
qu.1.1.editing=useHTML@
```

Flash Question Authoring

To create a Flash movie, you will need Flash development software (for example, Adobe® Flash® CS3) and some familiarity with Flash programming. Your flash movie must implement the following features.

- A pair of variables called **response** and **locked** should be declared.
- Whenever a student interacts with the movie and brings it to a new state, that state should be translated into a text string and assigned to the variable **response**. You are free to decide for yourself how this translation is done.
- When the Flash movie starts, it reads the variable response and finds a string that was returned by a previous call to GetVariable("response"). You must program the movie to initialize its appearance to the state consistent with that response.
- The Flash movie also reads the variable **locked**. This variable determines when a student is allowed to modify his/her response. When the value of locked is equal to true, this signals to the Flash movie that the question is in 'feedback' mode; in this case, the movie should present in a form suitable for feedback.

After creating the Flash movie, upload it to a folder on the class web site, as described in *Flash Movies and Java Applets in Questions (page 89)*

Finally, you need to host the Flash movie in a Maple T.A. question, using the Flash question type. Flash questions must be authored in a plain text script file.

For example, the demonstrated code above is for a Flash question which asks students to move the disk to a specific point on the Cartesian plane.

9.8 Free Response and Fill-in-the-Blank Questions

Free response questions present students with a question and a response object (or entry cell) in which to express their response. Upon grading, the student response is evaluated against the correct answer defined for the question.

The system provides many styles of automatically graded free response question types that differ in terms of presentation, accepted response types, and the graders applied to student responses. The range of question types includes questions that accept and grade symbolic and numeric mathematics, chemistry expressions, and a variety of text-based responses.

List Questions and Blanks Questions

There are two types of **Text-based** free response (or fill-in-the-blank) questions in the system. Each evaluates student responses by comparing the student response text string to that of the correct answer (as coded into the question).

- **list questions**
- **blanks questions**

Input Style and Options

1. Inline Blanks

Both **List** and **Blanks** questions allow you to display free response entry cell *blanks* inside the HTML question statement. This method of presentation can provide *inline* student response cells anywhere within the question.

The question in **Figure 9.10** demonstrates the power and flexibility of the question designer question type. There are 6 individual response cells included, each of which requires a different type of student response.

Notes:

- Response cells can appear anywhere within the formatted question statement, including within tables.
- Response cells can consist of any individual question mode, including response cells that handle text (lenient or stringent string matching), numbers, formulas, or any other question variety.
- Partial credit can be allowed for exact-graded text or menu-style entry cells.

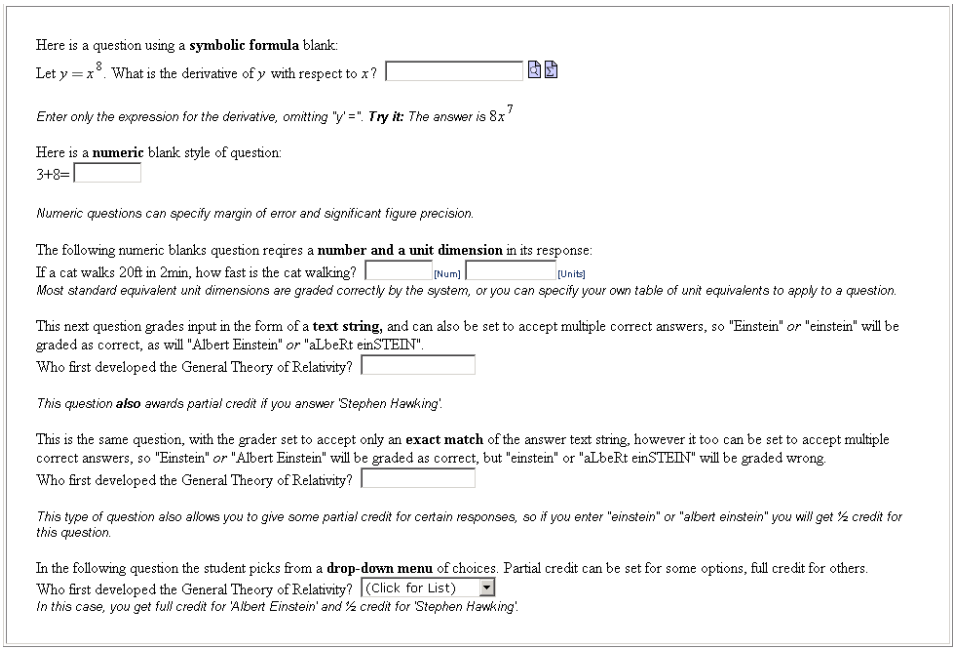

Here is a question using a **symbolic formula** blank:
Let $y = x^8$. What is the derivative of y with respect to x?
*Enter only the expression for the derivative, omitting "y' =". **Try it:** The answer is $8x^7$*
Here is a **numeric** blank style of question:
3+8=
Numeric questions can specify margin of error and significant figure precision.
The following numeric blanks question reqires a **number and a unit dimension** in its response:
If a cat walks 20ft in 2min, how fast is the cat walking? [Num] [Units]
Most standard equivalent unit dimensions are graded correctly by the system, or you can specify your own table of unit equivalents to apply to a question.
This next question grades input in the form of a **text string,** and can also be set to accept multiple correct answers, so "Einstein" *or* "einstein" will be graded as correct, as will "Albert Einstein" *or* "aLbeRt einSTEIN".
Who first developed the General Theory of Relativity?
*This question **also** awards partial credit if you answer 'Stephen Hawking'.*
This is the same question, with the grader set to accept only an **exact match** of the answer text string, however it too can be set to accept multiple correct answers, so "Einstein" *or* "Albert Einstein" will be graded as correct, but "einstein" or "aLbeRt einSTEIN" will be graded wrong.
Who first developed the General Theory of Relativity?
This type of question also allows you to give some partial credit for certain responses, so if you enter "einstein" or "albert einstein" you will get ½ credit for this question.
In the following question the student picks from a **drop-down menu** of choices. Partial credit can be set for some options, full credit for others.
Who first developed the General Theory of Relativity? (Click for List)
In this case, you get full credit for 'Albert Einstein' and ½ credit for 'Stephen Hawking'.

Figure 9.10: Demonstration of Various Free-Response and Fill-in-the-Blanks Questions

Both **List** questions and **Blanks** questions can be set to require students to enter text (textbox style presentation), or to select from a list of entries (menu-style presentation). Each style has its advantages.

- Use textbook style presentation when you want to evaluate short or long text responses in true *fill-in-the-blank* style. You can set the grader to **exact** and enforce an *exact match* between the student response and the correct answer, or you can set it to **relaxed**, a more lenient grading style in which capitalization and punctuation are ignored.
- Alternatively, you may want to constrain student input and test general knowledge by providing a longer excerpt with isolated phrases blanked out, but using the menu style presentation to require that a student select from a drop-down list of possible answers.

2. Traditional Blanks Presentation

Because you control the HTML formatting inside the question as well as the position of the blanks inside a List or Blanks question, you can also use those question types to present a response cell on a separate line by inserting paragraph or line breaks inside the question statement.

Comparison of Question Types Supported

The **List** question offers the widest variety of response objects.

Question Mode	Text - Exact Grading	Text - Relaxed Grading	Text - Menu	Formula	Numeric	Multiple Choice	Multiple Selection
List Question	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Blanks Question	**Yes**	**Yes**	**Yes**	**Yes**	**No**	**No**	**No**

Comparison of List Questions vs. Blanks Questions

List and **Blanks** questions are very similar; however, **List** questions provide: a greater diversity of graders that can be applied, better control over the specification of multiple correct answers, and the ability to return partial credit for answers that are not 100% correct.

Authoring Free Response Questions

The easiest way to create **Blanks** free response questions is by using the Question Editor, but you can also author **List** questions using the LaTeX authoring or plain-text scripting methods.

Key Word or Phrase Questions

Although similar to these question types, key word or phrase questions use a different grading approach for evaluating student responses, in which student responses are evaluated for the presence of an identified word or phrase *anywhere* within the student response.

See Also:

Blanks Questions (page 153)

Free Response (Fill-in-the-Blank) Questions in the Question Editor (page 95)

Palette Questions (page 165)

Key Word or Phrase Questions (page 160)

List Questions (page 160)

Math Question Types - Overview (page 132)

Short Phrase Questions (page 177)

Fill-in-the-Blank Question Example Script (page 289)

Rules for Grading Free Response Questions

By default, text-based free response question types, including **List Questions**, **Blanks Questions**, and **Short Phrase** questions apply standard rules when grading student responses. Standard grading behavior applies the **relaxed** grader style.

Relaxed Grader Rules

The process for evaluating student responses against the correct *answer* is as follows. With the student response *and* correct answer:

- Change uppercase Roman letters to lowercase Roman letters.
- Remove punctuation characters.
- Do not change lowercase Roman letters or numbers (0 - 9).
- Replace all other characters with the ASCII space character.
- Convert strings of spaces to a single space character.
- Compare the modified student response to the **answer** field string indicated in the question. Spaces in the response must correspond to spaces in the **answer** value.

For example, if the **answer** field value is **"Albert Einstein"**, then the following responses are graded correct.

- **"Albert Einstein"**
  "albert einstein"

- **"a'lbert Ein-stein"**

However

- **"A l b e r t Einstein"**

is incorrect (as is **"Aleister"**, of course).

Note: The relaxed grader rules are not appropriate for international languages since non-Roman characters are removed.

Exact Grader Rules

The optional **exact** grader style that can be specified in **List** questions and **Blanks** questions modifies the above rules to recognize and enforce capitalization, punctuation, and other characters but still applies the above algorithm for handling spaces.

Again, assuming the answer field value is **"Albert Einstein"**, then the following answer is graded correct:

- `"Albert Einstein"`

However,

- `"aLBert eiNsteiN"  or "albert einstein" or "a'lbert Ein-stein"`

is graded wrong.

Palette Grader Rules

The **Palette** question type is intended for use in chemistry content areas, and applies a modified exact grader designed to recognize and enforce superscripts and subscripts as well as the order of certain superscript and subscript operators.

Math Grader Rules

Each of the **Formula** question varieties, as well as the **Numeric** question type apply question-specific grading rules, and support methods for setting margin of error and tolerance, and equivalent unit dimensions at the level of the question.

See Also:

List Questions (page 160)

Free Response and Fill-in-the-Blank Questions (page 156)

Key Word or Phrase Questions (page 160)

Palette Questions (page 165)

Short Phrase Questions (page 177)

Setting Answer Tolerance in Numeric Questions (page 86)

9.9 Key Word or Phrase Questions

Question Mode: key words

```
qu.x.y.mode=key words@
```

The **key words** or phrase questions search the student's response for specified key words or phrases but ignore other text. This question type is not supported by the **Question Editor** in this release. You must author **key words** questions using scripts or LaTeX.

See Also:

Key Word or Phrase Question Example Script (page 291)

Free Response Text-based Questions in LaTeX (page 253)

9.10 List Questions

List questions are similar to **Blanks** *menu* style and **Blanks** *text* style question types in terms of their application, except they only allow you to set a single response object in the question. They also offer a few major enhancements.

- You can define multiple correct (and even incorrect but expected) answers.
- For each specified answer, you can control the amount of credit awarded for each specified answer.
- For each specified incorrect answer, you can define response-specific comments (feedback).
- You can define the display style for the student response input cells - either **text** or **menu** style display for student response objects (entry cells).
- Graders can be set to **exact** or the more lenient **relaxed** style.

Summary of Question Behaviors

Graders

- default grader: "exact"
- option: "relaxed" (set grader=relaxed)
- option: "regex" (set grader=regex)
  (This provides access to the student response set as a regular expression, for interacting with custom graders.)

Input Display Style

- default display: "textbox"
- option: "menu"
  To change the order of menu entries, set display.permute = false

Sizing of Textbox Blanks

- blanks sizing: based on max of answer length

Full and Partial Credit

- at least one answer field must give full credit
- all other answers can be set for credit from 0-1 (specified in decimal numbers)

Grading Algorithm

Grading is done by comparing the student response with each answer in order until a match is found, and the corresponding credit is then returned. If no match is found, the credit is zero. The emphasis on comparing in order is necessary

because someone writing regular expression questions might start with a very specific criteria for full credit and then allow more vague criteria for low credit.

Updating **Example 1** on List Question Example Script, if the answers and credits are changed to:

```
qu.1.1.answer.1=(E|e)instein@
qu.1.1.credit.1=1.0@
qu.1.1.answer.2=(E|e).*@
qu.1.1.credit.2=0.1@
```

then "Einstein" and "einstein" receive full credit and any other response beginning with "E" or "e" receives 10%. If the order of the two criteria is reversed, it is impossible to receive full credit.

See Also:

Rules for Grading Free Response Questions (page 158)

Free Response and Fill-in-the-Blank Questions (page 156)

List Question Example Script (page 292)

9.11 Matching Questions

Question Mode: Matching

```
qu.x.y.mode=Matching@
```

Matching questions display two lists. A student must match each element of the first list with an element in the second list.

Matching question types allow only 1 to 1 matches in the correct answer. Matches of 1 element to many are not available in the system. As an alternative, consider using Multiple Choice or Multiple Selection questions.

The elements can be HTML-compatible reference objects, for example, images.

The system displays the elements of the first list in rows of 3, by default. Each element has a drop box that contains numbers relating to the elements from the second list, which are listed below.

Example

The following algorithmically generated **Matching** question uses images (not text) for matching pairs.

```
qu.1.1.mode=Matching@
qu.1.1.name=pKa sort by matching@
qu.1.1.editing=useHTML@
qu.1.1.algorithm=
$a=switch(rint(4),
"<img src=../classes/myclass/pka/pka01a01.gif>",
"<img src=../classes/myclass/pka/pka01a02.gif>",
"<img src=../classes/myclass/pka/pka01a03.gif>",
"<img src=../classes/myclass/pka/pka01a04.gif>");
$b=switch(rint(4),
"<img src=../classes/myclass/pka/pka02a01.gif>",
"<img src=../classes/myclass/pka/pka02a02.gif>",
"<img src=../classes/myclass/pka/pka02a03.gif>",
"<img src=../classes/myclass/pka/pka02a04.gif>");
$c=switch(rint(5),
"<img src=../classes/myclass/pka/pka03a01.gif>",
"<img src=../classes/myclass/pka/pka03a02.gif>",
"<img src=../classes/myclass/pka/pka03a03.gif>",
"<img src=../classes/myclass/pka/pka03a04.gif>",
"<img src=../classes/myclass/pka/pka03a05.gif>");
$d=switch(rint(6),
```

```
"<img src=../classes/myclass/pka/pka04a01.gif>",
"<img src=../classes/myclass/pka/pka04a02.gif>",
"<img src=../classes/myclass/pka/pka04a03.gif>",
"<img src=../classes/myclass/pka/pka04a04.gif>",
"<img src=../classes/myclass/pka/pka04a05.gif>",
"<img src=../classes/myclass/pka/pka04a06.gif>");
$e=switch(rint(2),
"<img src=../classes/myclass/pka/pka05a01.gif>",
"<img src=../classes/myclass/pka/pka05a02.gif>");
@
qu.1.1.format.columns=5@
qu.1.1.question=Sort the molecules below from highest to lowest pKa:@
qu.1.1.term.1=1 (lowest pKa, Most acidic)@
qu.1.1.term.1.def.1=$a@
qu.1.1.term.2=2@
qu.1.1.term.2.def.1=$b@
qu.1.1.term.3=3@
qu.1.1.term.3.def.1=$c@
qu.1.1.term.4=4@
qu.1.1.term.4.def.1=$d@
qu.1.1.term.5=5 (highest pKa, Least acidic)@
qu.1.1.term.5.def.1=$e@
```

See Also:

Matching Questions in the Question Editor (page 98)

Multiple Choice Questions (page 163)

Multiple Selection Questions (page 164)

Question Types Available in the System (page 131)

9.12 Multipart Questions

Multipart Questions

Question Mode: multipart

```
qu.x.y.mode=multipart@
```

A **multipart** question can contain a variety of related question materials, data, even case scenarios for different types of assessment and learning.

Although the **multipart** mode is a question type, its component question parts can be of any question type, including multipart. This allows nested multipart questions.

The **Question Editor** is the recommended method for authoring multipart questions. However, you can also use the plain-text script file path or the LaTeX authoring path.

See Also:

Multipart Questions in the Question Editor (page 100)

Multipart Question Example Script (page 295)

Question Designer Questions (page 166)

Tutorial: Working with Multipart Questions (page 112)

Multipart Questions - Adding or Editing Parts

It is recommended that you organize and edit question parts in a topic, and then build the topic into multipart question only after you are finished. For details on editing multipart questions, see *Tutorial: Working with Multipart Questions (page 112)* .

Multipart Questions: Partial Credit

Assigning partial credit in multipart questions is closely related to establishing point values for the individual parts during construction of the question in the **Question Editor**. As you build a multipart question in the **Question Editor**, you can assign a question weight to each part of the question. In this way, you control partial credit in multipart questions.

See Also:

Multipart Questions in the Question Editor (page 100)

Partial Credit and Weighted Scoring (page 113)

Tutorial: Working with Multipart Questions (page 112)

Multipart Questions - Sequencing Parts

When creating a multipart question, identify the source question for each part.

- To select questions as *parts*, you can add them individually or select multiple questions from a single topic (using the check boxes beside the questions).
- If you select multiple questions from the same topic during the same authoring session, the multipart question-authoring tool in the **Question Editor** inserts each question part *in the same order in which it appears in the source topic*.

See Also:

Multipart Questions in the Question Editor (page 100)

9.13 Multiple Choice Questions

Question Mode: multiple choice; non permuting multiple choice

```
qu.x.y.mode=multiple choice@
qu.x.y.mode=non permuting multiple choice@
```

Multiple choice questions in the system can have any number of choices. You can also add incorrect choices after questions are initially created.

When making assignments, you choose to permute the choices with each delivery (**multiple choice**) or maintain a static choice order (**non permuting multiple choice**).

You can provide response-specific feedback. That is, you can provide a distinct comment for each choice. For every choice in your question, add a **comment** line of the form:

```
qu.x.y.comment.<n>=<comment_text>@
```

where **<n>** is the same positive integer as in the corresponding choice line:

```
qu.x.y.choice.<n>=<choice_n>@
```

To enter response-specific feedback, use a text or HTML editor, LaTeX, or the **Question Editor** Edit Source functionality.

See Also:

Multiple Choice Questions in the Question Editor (page 101)

Multiple Choice Question Example Script (page 296)

9.14 Multiple Selection Questions

Question Mode: multiple selection; non permuting multiple selection

```
qu.x.y.mode=multiple selection@
qu.x.y.mode=non permuting multiple selection@
```

Multiple selection questions are similar to multiple choice items. However, they can require the user to identify more than one correct answer. You can create as many choices as you want.

In **multiple selection** questions, the selections are permuted each time they appear. In **non permuting multiple selection** questions, the selection order is static.

Display of Choices in Multiple Selection Questions

By default, the system presents choices in a stacked, vertical list. If you want to set choices horizontally, use the **Question Designer** question type, and set your response object to **multiple selection** or **non permuting multiple selection**. For more information, see *Question Designer Questions (page 166)*.

Grading Multiple Selection Questions

A multiple selection question is graded fully correct if the student selects all the correct answers (and no incorrect answers). The grader applies negative credit for selecting incorrect answers. The score is calculated using the following algorithm.

1. Score +1 for each selected correct answer and -1 for each selected incorrect answer.
2. Sum these scores. If this result is negative, then it is replaced by 0.
3. Divide by the total number of correct answers.
4. Multiply by the total points for the question defined during assignment creation.

That is, for a multiple selection question the score is given by the formula

$$\frac{\text{\# selected correct answers - \# selected incorrect answers}}{\text{Total \# of correct answers}} \times \text{Total \# of points}$$

with the exception that if the numerator is less than 0, it is set to 0.

For example, if the correct answers are (a) and (b), and the incorrect answers are (c) and (d), then selecting all four would score 0%, and selecting a, b, and c would score 50%.

If the multiple selection question is included as a part of a multipart or question designer question, the question score is multiplied by the question part weighting to determine the grade for the question. Question part weighting is set in the question definition.

See Also:

Algorithmic Multiple Selection Question Example Script (page 284)

Multiple Selection Questions in the Question Editor (page 101)

Multiple Selection Question Example Script (page 296)

9.15 Palette Questions

Question Mode: palette

```
qu.x.y.mode=palette@
```

Palette questions enable you to create your own customized palette for student response entries.

- Before creating a new palette definition, you should get familiar with the Equation Editor requirements and syntax.
- To add a new palette definition, you need to create a question bank in plain-text script format and add the toolbar definition at the beginning of the file using the **qu.env.palette.toolbarname=** structure (see below). You can add one or more toolbar definitions at a time. You can add one or more palette entries to a toolbar definition.
- After importing the question bank, you will add a palette question using the Question Editor and select from the list of available palettes you created in the Palette Style drop-down menu.

Note: Palette questions should only be created using the **Question Editor**. Palette questions work by using an Equation Editor to generate Presentation MathML for the student response and compare it with the Presentation MathML of the answer. However, because there are many different ways of making Presentation MathML for the same (or almost same) marked-up expression, creating the expressions outside the Equation Editor may generate unpredictable results.

Grading of Palette Questions

The default behavior for grading palette question student responses is to perform string-matching on the palette-created expression.

Example Toolbar Definition:

Provide a toolbar definition by including

```
qu.env.palette.toolbarname=
  <toolbar def>
@
```

Example Palette Definition:

Provide palette definition(s) by including

```
qu.env.palette.subsup=
  <btn>FRAC</btn>
  <btn>SQRT</btn>
  <btn>SUB</btn>
  <btn>SUP</btn>
@
qu.env.palette.greek=
  <tb>
    <btn>SYMBOLPALETTE</btn>
  <sub cols=3>
    <btn>alpha</btn>
    <btn>beta</btn>
    <btn>gamma</btn>
  </sub>
  </tb>
@
```

Example Question

```
qu.1.1.mode=palette@
qu.1.1.question=
  <the text of the question>
@
qu.1.1.answer=
  <the text of the answer; this is in MathML and is intended to be editable>
@
qu.1.1.toolbar=<ref to external toolbar definition>@
```

See Also:

Palette-based Symbolic Editor Questions in the Question Editor (page 103)

Equation Editor Requirements and Syntax (page 192)

Rules for Grading Free Response Questions (page 158)

9.16 Question Designer Questions

Question Designer questions are a more flexible and extensible form of free response question that can include multiple response areas. In this way, they are similar to multipart questions in that they provide a shell structure in which to embed other questions. Question Designer questions support the following features.

- Using the Question Designer, you can enter question text and the correct answers wherever they appear.
- Questions can contain multiple response cells, for example, **Fill-in-the-Blanks** questions
- The widest variety of types of response cells (linked to specific question grader types) anywhere in the body of a question

Question types supported in the Question Designer:

- All Formula Question Types: Formula, Formula Mod C, Restricted Formula, Equation, MultiFormula, ntuple, Formula List, Chemical Equation
- Numeric Question Types: with and without required unit dimensions
- Multiple Choice, Non-permuting Multiple Choice (displayed vertically or horizontally)
- Multiple Selection, Non-permuting Multiple Selection (displayed vertically or horizontally)
- List Questions, including text or menu input styles, exact or relaxed graders, multiple correct or incorrect responses can be specified, partial credit can be defined for each response
- Maple-graded: Formula and Maple Syntax
- Essay

As in a multipart question, each subpart of the Question Designer question must itself be a well-formed question. The only exception is that when used to define a blank inside a Question Designer question, any of the above question types may omit the **question** field (because this is actually defined *inline* in the body of the larger question statement).

The **Question Designer** question type is a *shell* that is internally consistent with other questions. Numeric blanks inside Question Designer questions, for example, will grade the same as single-part *numeric* questions because the grading is actually done by the numeric question itself. Likewise, each embedded question type inside a Question Designer question supports all the features offered with that individual question type (for example, margin of error, significant figures, and tolerance).

Note: The Question Designer question type supersedes the Inline question type.

See Also:

Question Designer Question Example using Various Embedded Questions (page 298)

List Questions (page 160)

Authoring with the Question Designer

The **Question Designer** is a powerful and flexible question container that can hold one or more graded answer regions or **Response Areas.** A variety of grading behaviors can be combined and placed anywhere within the formatted html that defines the question. For example, questions can be placed inside tables, formatted text, bulleted lists, and can include embedded images or media objects.

In addition to flexible question and response areas, it provides an environment similar to a word processor. See **Figure 9.11**. Apply character formats such as bold, italics, and fonts using toolbar buttons. You can insert images, tables, or links to other file resources.When you are satisfied with the appearance of the question, insert answer regions and then apply grading and other behaviors.

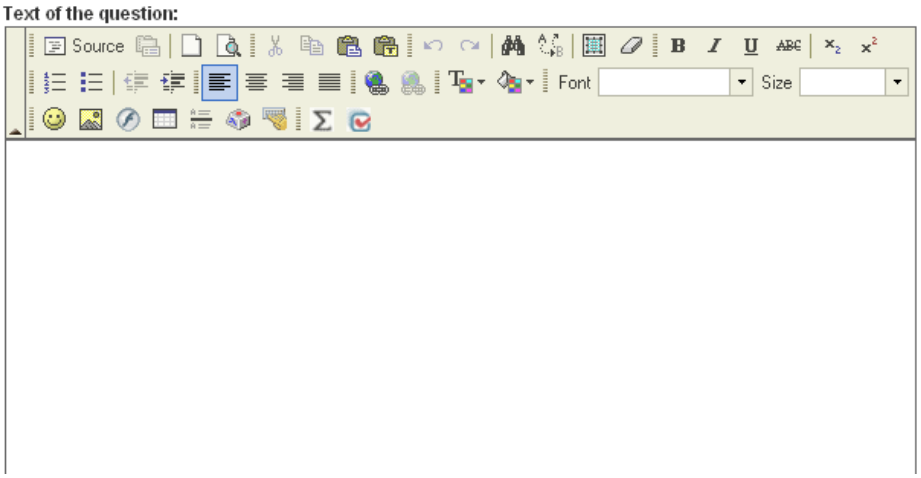


Figure 9.11: Question Designer Screen

Features

Questions authored in the Question Designer can contain:

- Any number of answer regions. Parts of your question can require text, numeric or formula answers while other parts can require multiple choice, essay answers—all in a single question.
- Standard HTML elements. You can include character formatting, HTML tables, embedded images and media objects.
- Any number of images, tables, algorithmic variables and inline math expressions as required.

From the Question Designer screen, you can view, add, or edit feedback, algorithm, information, hint, and solution fields.

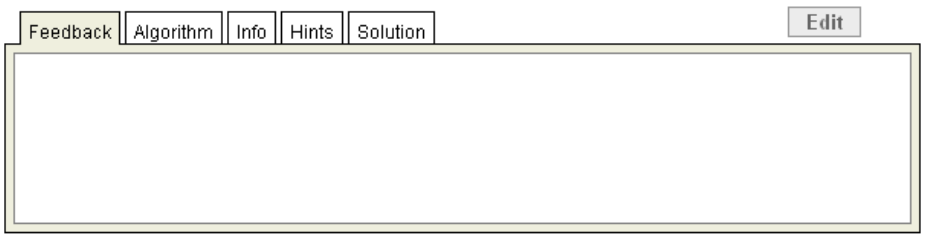


Figure 9.12: Tabs for Specifying Feedback, Algorithms, and Additional Question Info

See Also:

Question Types and Properties (page 169)

Edit/Insert Response Area (page 170)

Formatting Tips

The **Question Designer** makes it easy to apply formatting to your question in many of the same ways you use your word processor. There are a few limitations and requirements unique to the editor. These tips are offered to help you achieve the best results.

As with other web content, you may observe differences in the display of your question in various operating systems and browsers.

Character Formatting

- Use the toolbar buttons to apply character formatting.
- HTML tags are allowed and will be rendered as HTML

Math expressions in questions

- Use the Equation Editor to enter expressions. Click the **Sigma** icon to launch the Equation Editor dialog, as shown in **Figure 9.13**.
- Right-click (**Control**-click, Macintosh) in the input field to display palettes. See **Figure 9.14**.

Figure 9.13: Access the Equation Editor in the Question Designer

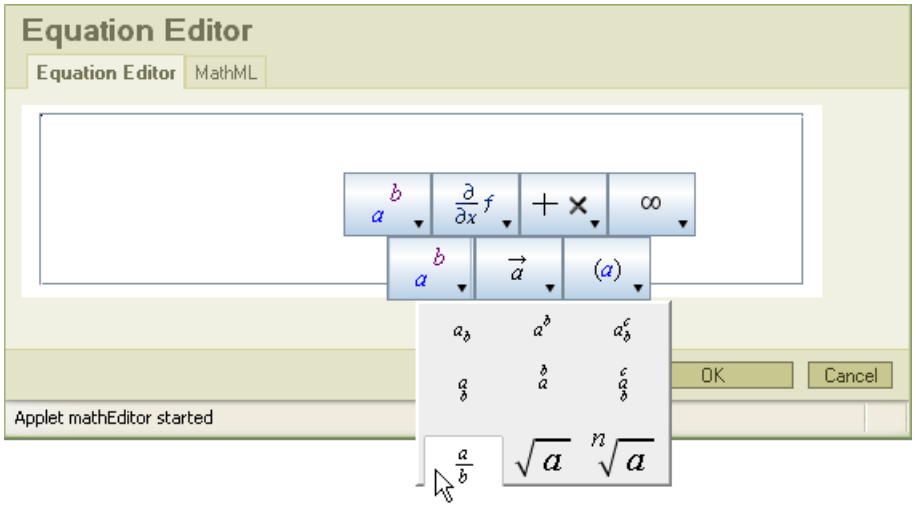


Figure 9.14: Equation Editor Palettes

Algorithmic Variables in questions

- Insert variables by typing their name. Use the required "**$**" at the beginning of the variable name: $variable.

Displaying the $ sign and the @ symbol in questions

- Insert the "$" sign by using the escape sequence "\", as follows: \$.
- Insert the "@" symbol by using the escape sequence "\" as follows: \@.

Important: When saving elements, remember to click **Finish** to save changes. If you change a value by simply typing content and do not click Finish, your changes will be lost.

Question Types and Properties

Questions can hold one or more **Response Areas**. You define the question mode, display style, grading behaviors, and correct (and incorrect) answers using the panel.

Supported Question Types

Essay (Ungraded)

Essay question **Response Areas** can be defined anywhere in the **Text of the question** panel.

Formula (Mathematical)

Formula questions provide full access to the formula mode math grading behaviors, including configurable grading of equivalent symbolic algebraic and equation expressions. The sub-modes are:

- Formula, for example, e^x sin(x^2)
- Formula that matches responses to within +C
- Vector of formulas, for example, (2t, t+1, 3t^2)
- Equation, for example, x +3y=1
- Formula without simplification, for example, 4x + x
- Chemical Equation, for example, 2H_2+O_2-->2H_2O

List Questions (Fill-in-the-blank or select correct answer from drop-down menu)

List questions are used to define fill-in-the-blank or select the correct answer from a drop-down menu questions. They can include multiple correct, incorrect, or partially correct responses.

Maple Questions

Maple questions include facilities for algebra, calculus, differential equations, discrete mathematics, graphics, numerical computation, and many other areas of mathematics.

You also have access to the plotting capabilities of Maple. You can plot a student response (or a function derived from a student response, for example, the definite integral of the student response) for a Maple-graded question type or display a plot for any question type. You can also use functions and routines that are contained in a separate Maple Repository (Maple Library). You can access your personal Maple library archives from within your Maple questions.

Multiple Choice and Multiple Selection

Multiple choice (one answer) and multiple selection (more than one answer) questions can be defined as either permuting (choices are shuffled) or non-permuting (order of choices is fixed as authored). Answer choices can be displayed horizontally or vertically.

Numeric

Numeric questions can be defined to require a number and a unit dimension or simply a number as the answer. You can set grading tolerance and answer precision using one of four methods.

The following sections explain how to use each of these question types within the response area.

Edit/Insert Response Area

Response Areas are the areas of your question that hold the answer definitions. Using the **Edit/Insert Response Area** button, you activate and define response areas in your question.

To define answer regions in your questions:

1. Click the **Insert/Edit Response Area** button ( ).

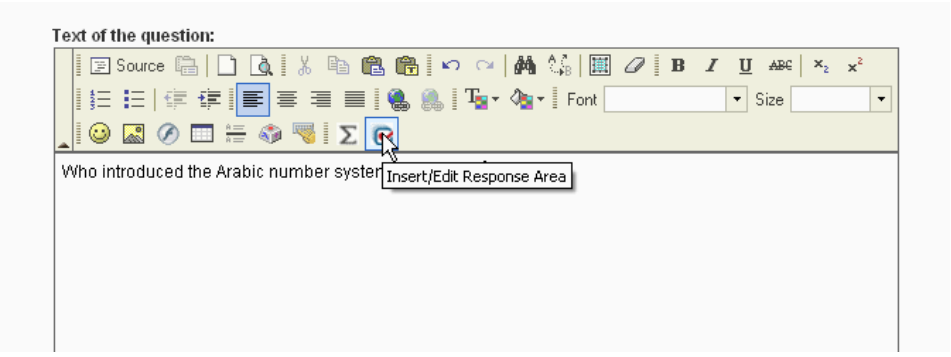
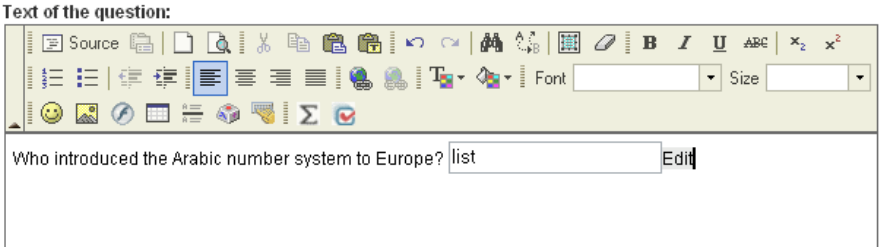


Figure 9.15: Add Response Area in the Question Designer

2. The **Edit Response Area** dialog opens. You can select from Formula, Maple, Multiple Choice, Numeric, List, or Essay question types.

3. Define the properties for the question type and click **OK**. See the next section for details on specifying these properties.
4. The response area is now displayed in the **Text of the question** panel.



Sample Question in the Question Designer with a List Response Area

See Also:

Authoring with the Question Designer (page 167)

Question Types and Properties (page 169)

Response Area Properties

Essay Question

To create an Essay question:

1. Enter the essay question in the **Text of the question** panel.
2. Click **Insert/Edit Response Area.**
3. Select the **Essay** question type in the **Edit Response Area** dialog.
4. Indicate the number of rows and columns (length) of the response area for the essay question and click **OK**.
   - You can define the size of the essay blank using rows and columns. A typical screen-width size is 65-70 columns wide. Rows directly correspond to the lines of text you want to provide in a typical response.
   - If a student enters more text than will fit in the preconfigured text window, the cell will automatically insert a scroll bar and allow the student to continue his or her response.
5. Click **OK** and then **Finish** to save and preview the question.

Formula

To create a Formula question:

1. Enter a question in the **Text of the question** panel. Optionally, enter the question using symbolic math by clicking the **Equation Editor** icon. (The **Equation Editor** opens. See *Formatting Tips (page 168)* .)
2. Click **Insert/Edit Response Area.**
3. Select the **Formula** question type in the **Edit Response Area** dialog.
4. Select a formula math grader by clicking **Sub-type** drop-down list.
5. Enter the answer in the **Answer** field.
6. Click **OK** and then **Finish** to save and preview the question.

List Question

To create a List question:

1. Enter a question in the **Text of the question** panel.
2. Click **Insert/Edit Response Area.**

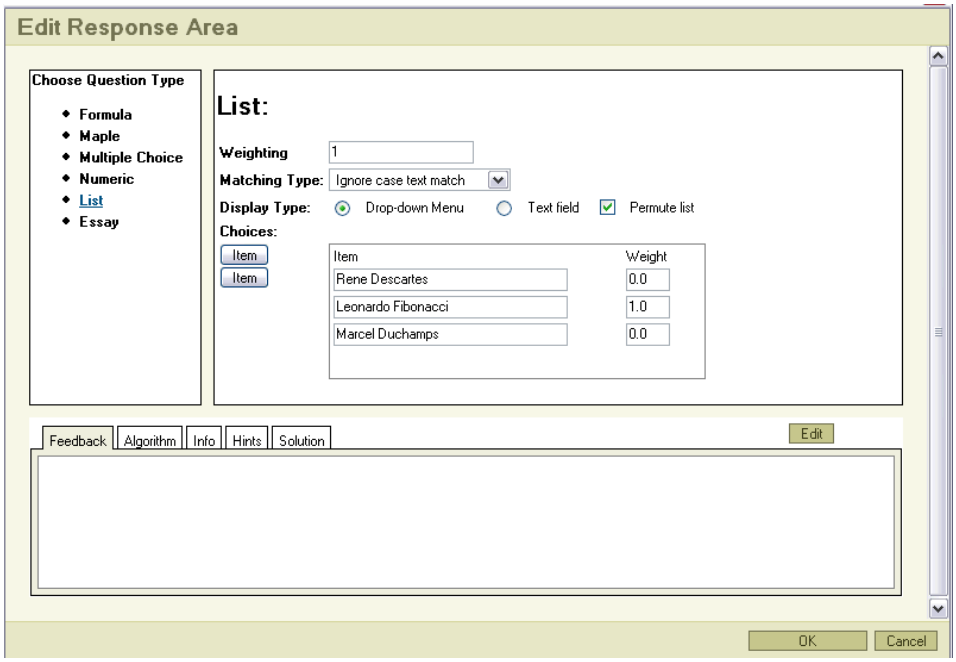


Figure 9.16: List Response Area in the Question Designer

3. Select the **List** question type in the **Edit Response Area** dialog.
4. Select the Display Type. The default **Display Type** is **Text field**, which presents students with a blank response area. To present students with a drop-down menu of choices, select **Drop-down menu**.
   - When the display type is Text field, you can set the grading standards. To apply strict grading, case-sensitive, and literal string-matching as your grading mode, set matching type to **Exact text match** from the **Matching Type** drop-down list. To invoke a less stringent grader that ignores case sensitivity and punctuation, select **Ignore Case text match.** You can also use a regular expression matching grader to customize the grading routine.
   - Enter answer choices in the **Item** fields. To add items, click the top **Item** button.
   - As you add expected answers the system assigns them a credit value of "0.0" in the right column. Edit the weights as desired, using 1 (fully correct) or any decimal value between 0 and 1 for partially correct answers.
5. Click **OK** and then **Finish** to save and preview the question.

See Also:

Rules for Grading Free Response Questions (page 158)

Maple

To create a Maple question:

1. Enter a question in the **Text of the question** panel.
2. Click **Insert/Edit Response Area**.
3. Select the **Maple-graded** question type in the **Edit Response Area** dialog.
4. Select an **Expression type: Formula** or **Maple Syntax.**
5. Using Maple code, enter the correct answer in the **Answer** field. For example, if the question is find the indefinite integral of sin(x), then the **Answer** field must contain the Maple code **int(sin(x),x);**
6. By default, the Maple code to grade the student answer is given: **evalb($ANSWER-$RESPONSE=0);**
7. Click **OK** and then **Finish** to save and preview the question.

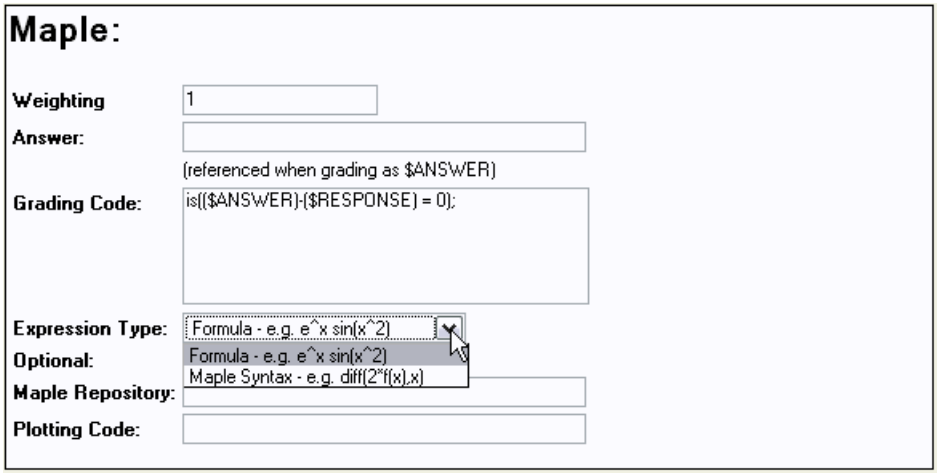


Figure 9.17: Maple-graded Response Area in the Question Designer

Options

- You can enter Maple code to plot the student answer. For example, enter an expression in x. The Maple grading code is **plot($RESPONSE, x=0..10);**
- You can access your personal Maple library archives from within your Maple questions. You must first upload your archive files into the class file space on the server. When you create a Maple-graded question, you specify the location of the archive as part of the question definition. When executing the Maple code associated with that question, the Maple libname variable is set automatically to access the archive files specified. Click the blank field in the **Maple repository** row and enter the Library location.

Multiple Choice

To create a Multiple Choice question:

1. Enter a question in the **Text of the question** panel.
2. Click **Insert/Edit Response Area.**
3. Select the **Multiple Choice** question type in the **Edit Response Area** dialog.

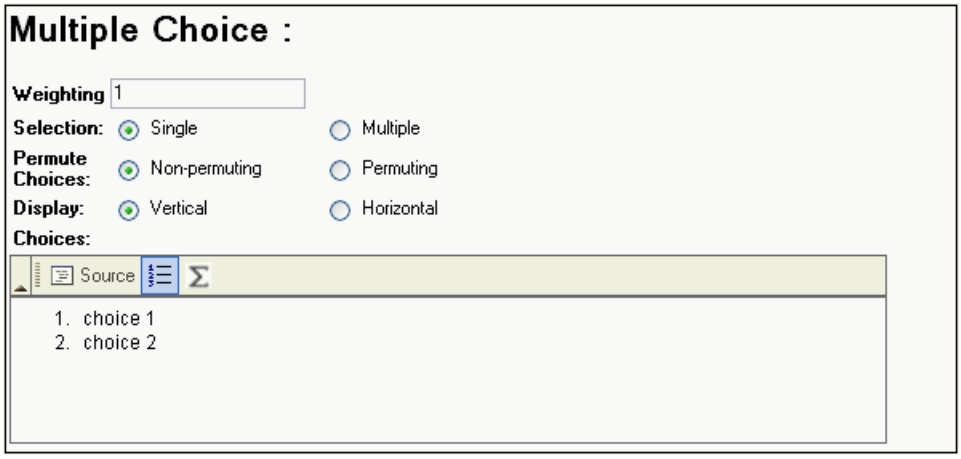
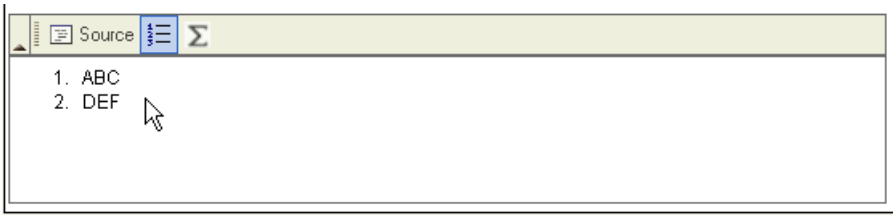


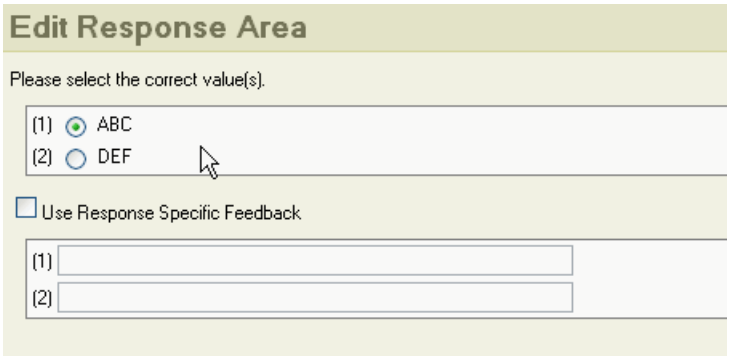
Figure 9.18: Multiple Choice Response Area in the Question Designer

4. Select **Single** for one correct answer or **Multiple** for multiple correct answers.
5. To rearrange the order of the displayed choices, select **Permuting**.
6. Select the display format of the choices in a table: **Vertical** or **Horizontal**.
7. In the **Choices:** area enter the text that defines the choices for the question.



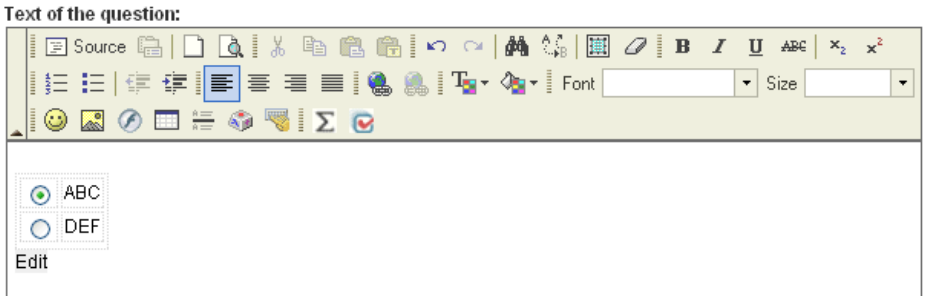
Define Choices for Multiple Choice

8. Click **OK**. The **Edit Response Area** dialog opens. Select the correct values.



Specify Correct Choice

9. Click **OK**.



Finish Multiple Choice Type in Question Designer

10. Click **Finish** to save and preview the question.

Numeric (with or without required unit dimensions)

To create a Numeric question:

1. Enter a question in the **Text of the question** panel.
2. Click **Insert/Edit Response Area.**
3. Select the **Numeric** question type in the **Edit Response Area** dialog.
   - Numeric questions can accept numbers without units as valid responses. The correct answer must be expressed as a number. The student response can be an expression (such as "1+2+3+4").
   - Numeric responses can be expressed in decimal form or scientific notation (for example, 3.24E4).
   - To require units, enter a unit in the **Units Part** field (for example, **m** for meters). If you do not enter a required unit dimension, the system displays only a single response cell that accepts only numbers. If you do enter a unit dimension, students are presented two cells, one for the number part and one for the unit part. The student must enter correct values in both cells to receive full credit.
   - Correct answers can be specified with a margin of error or range of tolerance.
   - Algorithmic variables can be created for use in the question statement, answer, hint, feedback, and solution fields.
   - The system automatically grades equivalent numeric expressions correct
4. Click **Finish** to save and preview the question.

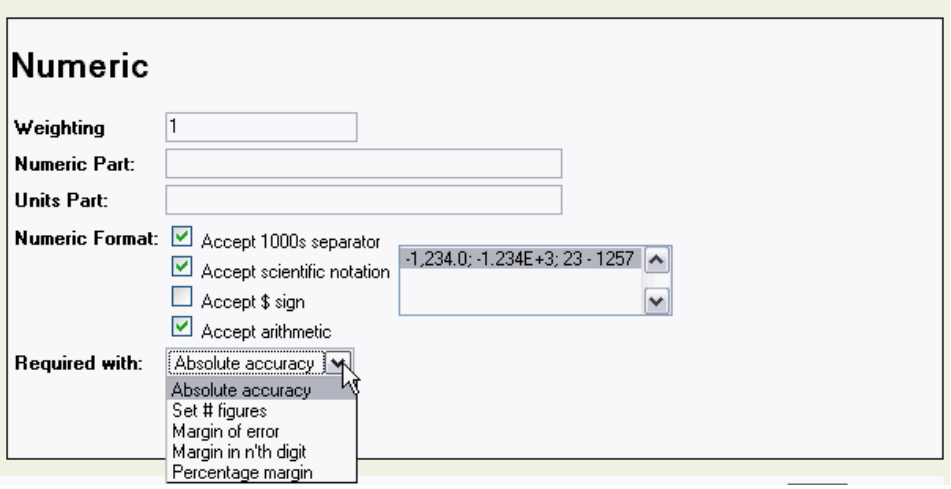


Figure 9.19: Numeric Response Area in the Question Designer

See Also:

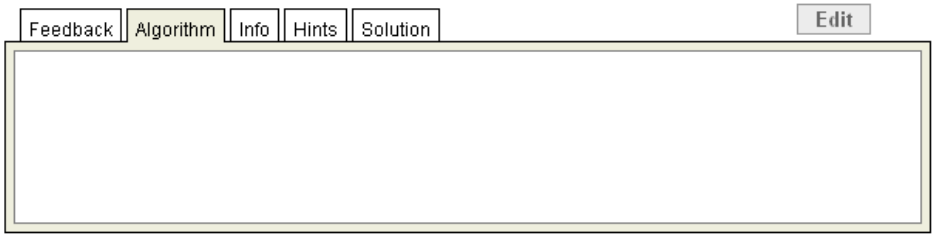
Authoring with the Question Designer (page 167)

Edit/Insert Response Area (page 170)

Question Types and Properties (page 169)

Insert Algorithm

Insert algorithms using the **Algorithm** tab in the **Question Designer** screen.



Algorithms Tab in Question Designer

Click the **Algorithm** tab in the **Question Designer** screen and then click **Edit** and follow the instructions on the screen.

Maple-generated plots to be included in questions are defined in the Algorithm Editor using the plotmaple command. When you define a plot in the Algorithm Editor, you can click **Refresh** to see the plot image, verify the plot statement is correct, and edit it if necessary. For more information on the Algorithm Editor, see *Adding and Editing Algorithms (page 78)*.

Note: When refreshing the algorithmic variable, you need to change the plot statement in order for the new image to appear. Changing a **plotdevice** or **plotoption** entry will not update the plot.

See Also:

Plotting Using Maple (page 201)

9.17 Short Phrase Questions

Question Mode: short phrase

```
qu.x.y.mode=short phrase@
```

Note: This question type is supported for backwards compatibility with earlier releases. It is recommended that you use the List question type instead.

Short phrase questions apply the same grading rules as other free response text questions. The question statement is presented first, followed by a student response entry cell on a separate line.

Short phrase questions can be authored only by using scripts or LaTeX. To ensure that this question type is appropriate, consider the following.

- To create a question with inline entry cells for student responses, use the blanks question type.
- To create a question for which the use of a specific word or phrase is important, but the overall structure of the response is not, use the key words question type.
- To create a question that is not automatically graded, use the essay question type.
- To create a graded free response math or science question, review the math/science question types supported by the system.

See Also:

Free Response and Fill-in-the-Blank Questions (page 156)

List Questions (page 160)

Rules for Grading Free Response Questions (page 158)

Short Phrase Question Example Script (page 300)

9.18 True-or-False Questions

Question Mode: True False

```
qu.x.y.mode=True False@
```

True-or-false questions are treated as a class of multiple choice question featuring only two choices.

See Also:

True-or-False Questions in the Question Editor (page 105)

Multiple Choice Questions (page 163)

True-or-False Question Example Script (page 300)

9.19 Advanced Question Types

The system supports the following applet question types. To create these questions, you must author the questions in LaTeX, or in a text or HTML editor.

- Dynamic figure label questions *Questions with Labeled Images (page 106)*, *Labeling Graphics in LaTeX Questions (page 263)*
- Graph-sketching questions *Graph Sketching Questions in LaTeX (page 260)*, *Graph Sketching Question Example Script (page 290)*
- Applet questions *Applet Questions (page 151)*

Note: While these question types cannot be created in the Question Editor, it is possible to embed an applet graph in a question that is created in the Question Editor, using the **Edit Source** functionality to edit the source code directly. For more information, see *Graph Plotting Applet in Questions (page 90)*.

See Also:

Questions with Plots (page 148)

Graph Plotting in LaTeX Questions (page 259)

Graph Plotting Applet (page 149)

10 Authoring Mathematical Questions

10.1 Authoring Mathematical Questions

Maple T.A. provides cross-platform support for learning content that includes numeric or symbolic mathematics expressions, and chemistry equations and expressions.

Fundamental Capabilities

- Integrated mathematics parser and Math Syntax for expressions
- Maple-graded expressions
- Graded free-response mathematics questions containing numeric responses, equations, formulas, physical units, and combinations of numbers, formulas, and units
- MathML support
- Inline expression rendering from graphing calculator syntax to formatted MathML expressions
- System evaluation of equivalent symbolic, numeric, equation, and unit expressions
- Automatic recognition of scientific notation in the answer field and in student responses
- Control over precision, significant digits, and margin of error/tolerance in questions
- Support for questions using Maple
- Support for questions that use plots generated by Maple

Mathematics Question Types

Maple T.A. offers 14 varieties of mathematical and scientific free response question types (including formula, numeric, and questions that use the Maple mathematical software to grade and plot student responses).

Symbolic and Equation Question Types

- *Maple-graded Formula (page 142)*
- *Maple Syntax (page 143)*
- *Formula (page 137)*
- *Restricted Formula (page 150)*
- *Formula with Units (page 139)*
- *Constants of Integration Questions (page 136)*
- *Equation (page 137)*
- *MultiFormula (page 145)*
- *Formula List (page 139)*
- *Vectors of Formulas (page 150)*
- *Graph Plotting in LaTeX Questions (page 259)*

Numeric Question Types

- *Numeric (page 146)* (including numbers with physical units)
- *Labeling Graphics in LaTeX Questions (page 263)*
- *Matrix (page 144)*

10.2 Basic Math Syntax in the System

- *Mathematical Expression Syntax (page 180)*
- *Variable Names Syntax (page 180)*
- *Operator and Function Syntax (page 181)*

Writing Mathematical Expressions

The basic syntax for entering mathematical formulas or expressions in the system enables you to quickly enter expressions using symbolic math. You can type formulas using standard mathematical notation (similar to that used for a graphing calculator) and, in general, the system correctly interprets it.

For example, the following formula is acceptable.

```
(x^2-2x+1) 2sin(x) (x^2+1)e^(-x^2)
```

Note: If a product includes one or more variables, always use an asterisk "*****". For example, specify **2*x*y** or **2*$A**.

For a **Maple** question, you must always include an asterisk (*).

```
(x^2-2*x+1)*2*sin(x)*(x^2+1)*e^(-x^2)
```

The most common mistake is to forget parentheses "**()**". For example, the expression:

```
1/(x+1)
```

is different from:

```
1/x+1
```

which the system interprets as:

$$\frac{1}{x} + 1$$

Alternatively, you can use MathML expressions, which are supported for both display and content within the system.

Variable Names

You can use any letter for a variable name.

Note: The grading system is case-sensitive with respect to variable names. So if the correct answer is:

```
(t+1)^2
```

then the response

```
T^2 + 2T +1
```

is graded wrong, but

```
t^2 + 2t +1
```

is correct.

Note: Be certain that the case of letters like **x** and **X** or **v** and **V** in answers match the case of the variables in questions.

Operator and Function Syntax

For multiplication, use an asterisk "*****". Alternatively, simply write:

- Two letters with a space (for example, **x y**)
- A letter and a number with or without a space (for example, **2x** or **2 x**). Note that two letters without a space represents a single variable name.
- For Maple-graded questions, you must include an asterisk *****, for example **2*x** and not **2x**, **x*y** and not **xy**, or **x*y** and not **x y**, which will be marked as incorrect.

For:

- Exponentiation, use the caret "^".
- Euler's Constant, 2.718..., use the letter **e**. (The exponential function is **e^x**.) For Maple-graded questions, use the **exp** function: **exp(1)** and **exp(x)** for **e** and **e^x**.
- Scientific notation, use the upper case letter **E**. (For example, 0.078 is 7.8E-2.)
- The trigonometric constant, **3.141**..., use **pi**. For Maple-graded questions, use **Pi**. In the Equation Editor (MathML Editor), select Pi from the palette.
- Common mathematical functions, use the standard abbreviated names (for example, **sin**, **cos**, and **tan**).
- The arguments of trigonometric functions, remember they are measured in radians.
- The square root function, use **sqrt(x)**. (Alternatively, use **x^(1/2)** or **x^0.5**.) In Text entry mode the square root function is **sqrt(x)**. (Alternatively, you can enter **x^(1/2)** or **x^0.5**.) In the MathML Editor, select the square root symbol from the palette.
- The inverse trigonometric functions, use the standard abbreviated names (**arcsin(x)**, **arccos(x)**, and **arctan(x)**).

Note: The variable **e** should *not* be assigned as a global variable in **Maple** code. It can be used, however, as a local variable.

See Also:

Editing with the Equation Editor (page 188)

Mathematical Functions & Operations (page 204)

Maple-graded Question Type - Overview (page 140)

10.3 Understanding the Math Capabilities in the System

The Maple T.A. system has powerful question types and capabilities designed for technical and scientific educators. It grades responses that contain symbolic or numeric answers. There are many restrictions and extensions to the type of answers that instructors accept, which determine the variety of question types in the system.

Learn more about the system math capabilities.

- How are questions with algebraic and numeric math authored? *Authoring Mathematical Questions (page 179)*
- Why are there so many varieties of mathematics questions? *Math Question Types - Overview (page 132)*
- How can I control margin of error and tolerance? *Setting a Margin of Error in Non-numeric Questions (page 184)*
- Can the system display WYSIWYG mathematics? *Math Question Types - Overview (page 132)*
- Does the system accept "graphing calculator" style math entry? *Using Formula Syntax (page 188)*
- What syntax constraints are placed on the processing of math expressions? *Basic Math Syntax in the System (page 180)*

- What functions and operators are supported? *Mathematical Functions & Operations (page 204)*
- What units does the parser understand? *Default Table of Physical Unit Equivalents (page 205)*
- Does the system support MathML? *MathML in the Questions (page 198)*
- Does the system support LaTeX? *LaTeX Authoring Overview (page 247)*

10.4 Student Responses and Grading

This section provides detailed information related to grading of student responses.

Student Responses

On an assignment, students can opt to enter their responses in Symbol mode or Text mode. This setting applies to any question type that supports both modes. For other question types, a specialized input field is presented to the students. For more information on the input field for a specific question type, see the section on that question type in *Question Types (page 131)*. For Maple syntax questions, the instructor specifies Symbol mode or Text mode for responses in the question itself. For more information, see *Maple-graded Questions in the Question Editor (page 96)*.

Controlling Answer Format in Numeric Questions

Note: You can use the **Question Editor**'s on-screen dialog boxes to set the answer format. For more information, see *Setting Answer Format in Numeric Questions (page 85)*.

Student responses do not need to be formatted identically to your answer to be correct. You can specify notations that the student is allowed to use.

Number Style

Use the **numStyle** field to set the acceptable formatting for student responses. List the acceptable number styles (in a space-separated list) as the value.

The available number styles are:

- **thousands** - responses containing commas (**,**) as a separator are graded correct
- **scientific** - responses specified using scientific notation, for example, **2.0E2**, are graded correct
- **dollars** - responses containing a leading dollar sign (**$**) are graded correct
- **arithmetic** - responses specified using the arithmetic operators, that is, +, -, *, /, (), and ^, are graded correct

For example:

numStyle = thousands dollars@

allows students to correctly enter responses using comma (**,**) separators or leading dollar signs (**$**).

Negative Number Style

Use the **negStyle** field to set the acceptable formatting for *negative* student responses. Set **negStyle** to one of **minus**, **paren**, or **both**.

- **minus** - Negative responses must be entered using a negative sign (**-**).
- **paren** - Negative responses must be entered using parentheses (**()**).
- **both** - Negative responses can be entered using a negative sign (**-**) or parentheses (**()**). **Note:** If you set **numStyle** to include **arithmetic**, then you cannot allow parentheses to be used to indicate a negative response. That is, **negStyle** must be set to **minus**.

See Also:

Numeric (page 146)

Setting Answer Format in Numeric Questions (page 85)

Controlling Answer Tolerance

There is not general agreement on how to grade numeric answers with respect to the number of significant digits. For example, if the correct answer to **3** significant digits is **2.77**, some instructors grade the response **2.77021** incorrect. Other instructors grade **2.77021** correct because they think that the inclusion of extra digits does not make a response incorrect.

In Maple T.A. you control how numeric responses are graded. For example, you can:

- Accept only **2.77** (and not **2.77021**). For more information, see **Setting the Precision**.
- Accept all answers that agree with the correct answer to **3** (or another positive integer) significant figures (that is, any response between **2.765** and **2.775** is correct). For more information, see **Setting a Margin of Error**.

For Maple-graded questions, see *Significant Digits in Maple Questions (page 185)*.

Setting the Precision

For the **numeric** question type, you can specify that the student response exactly match the correct answer, or specify that in addition to matching the value of your correct answer the student response have a specified number of significant digits.

Student responses that are correct up to the specified number of significant digits receive 50% credit.

For information on setting the precision for a **numeric** question in the **Question Editor**, see *Setting Answer Tolerance in Numeric Questions (page 86)*. For related plain-text script file examples, see *Answer Tolerance Example Script (page 285)*.

For other formula questions, unless a margin of error is specified, the value of the student response must match that of the correct answer up to the tolerance specified by the **tolerance** field, which is by default is the system floating-point limit, **1.0E-9**.

Setting a Margin of Error

You can specify a margin of error in one of two ways:

- **Percentage: ± x%**. This requires the student response to be within a specified percentage of the correct answer.
- **Absolute: ± x**. This requires the student response to be within a specified number of the correct answer.

For information on setting a margin of error for a **numeric** question in the **Question Editor**, see *Setting Answer Tolerance in Numeric Questions (page 86)*.

For information on setting a margin of error for other question types, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

For related plain-text script file examples, see *Answer Tolerance Example Script (page 285)*.

Note: It is recommended that you develop a policy for the tolerance in responses at the beginning of a project, so that every numeric question uses similar policies.

See Also:

Tutorial: Setting Margin of Tolerance in Non-numeric Questions (page 215)

lsu(n, x) (page 226)

decimal(n, x), sig(n, x), int(x) (page 223)

Setting a Margin of Error in Non-numeric Questions

Note: Use the **Question Editor**'s on-screen dialog boxes for setting the margin of error in numeric and numeric-with-unit questions. For more information, see *Setting Answer Format in Numeric Questions (page 85)*.

For an introduction to the concept of *margin of error*, see *Controlling Answer Tolerance (page 183)*.

The exact answer is not always required. In Maple T.A., you can specify a margin of error, within which student responses are graded correct, for any mathematical question type. You can specify a margin of error for answers that are static or defined using algorithmic variables.

Use "**?**" in the **answer** field to set a margin of error. The **?** (**plus or minus**) operator represents "+/-" in ordinary math notation. You can specify an *absolute* tolerance. For example, to accept **2.5 +/- 0.25**, that is, any response between **2.25** and **2.75** (inclusive), use **2.5 ? 0.25**. You can also specify a *percentage* tolerance. To accept **5.12 +/- 5%**, that is, any response between **4.864** and **5.376** (inclusive), use **5.12(1 ? 0.05)**.

Note: The **?** operator cannot be used in an algorithmic variable definition (or other algorithmic statement).

For **numeric** and **Maple-graded** questions, you cannot use the "**?**" operator. For information on specifying a margin of error for a **numeric** question, see *Setting Answer Tolerance in Numeric Questions (page 86)*. To specify a margin of error in a **Maple-graded** question, use Maple code.

Consider the following when working with margins of error:

- Precedence of the **?** Operator: The **?** operator is of the same precedence as "+" and "**-**". For example, **2*3 ? 0.5** is the same as **6 ? 0.5** and **2*(3 ? 0.5)** is the same as **6 ? 1**.
- Units: Because expressions with units have different dimensions from pure numbers, you must specify a dimensioned margin of error for a dimensioned answer. To specify "2kg with a tolerance of 0.1kg", enter **2kg ? 0.1kg** or **(2 ? 0.1)kg**. To specify "50cm within 5%", enter **(1 ? 0.05)50cm**.

Important: The expression **50cm ? 2** is invalid. It is a *mixed expression.* The left-hand side has dimension length. The right-hand side is dimensionless (a pure number).

- The **?** operator is **not available** to students: A student cannot specify a margin of error. The above issues do not arise for students.

For more examples and information, see *Tutorial: Setting Margin of Tolerance in Non-numeric Questions (page 215)*.

See Also:

Answer Tolerance Example Script (page 285)

Significant Digits in Maple Questions

The Maple **Digits** environment variable controls the number of digits that Maple uses when calculating with software floating-point numbers.

- The default value of **Digits** is 10.
- The value of **Digits** is changed by using the assignment operator. For example, **Digits:= 20;**

Using the evalf Command to Set the Value of the Digits Variable

The parameter **n**, which can be given as a second argument **(evalf(expr, n))**, sets the value of **Digits** for a *single* computation.

To round the exact answer, assign the exact answer to a dummy variable, and then use the **evalf(expr,n)** command in the **evalb** command to compare the student response to the rounded answer.

Example

Calculate the quadratic mean of the following list of data: [695, 607, 511, 588, 710, 500, 682, 515, 709, 478, 497, 657, 652, 634, 488, 615, 714, 579, 589, 493, 584, 484, 503, 711, 622, 695]. The answer requires 5 significant digits.

1. First calculate the exact numeric answer.
2. Round it to the required number of significant digits. For example:

```
exact_answer:=evalf(Statistics[Mean](<695, 607,  511, 588, 710, 500, 682, 515, 709, 478, 497,
657, 652, 634, 488, 615,  714, 579, 589, 493, 584, 484, 503, 711, 622, 695>)):
rounded_answer:=evalf(exact_answer,5):
evalb($RESPONSE=rounded_answer);
```

Using Maple Code to Prevent Cheating in Maple Syntax Questions

As the author, you can verify whether a student is using a Maple command to calculate the answer.

The following example scripts illustrate code that *allows* and *prevents* potential cheating by the student using Maple commands.

Examples

The student is required to **factor(x^2-1)**. The expected correct response is **(x-1)*(x+1)**.

In this example question code, the student could easily use the Maple **factor** command to obtain the answer.

```
question=Factor x^2-1.@
maple=evalb($RESPONSE=factor(x^2-1))@
type=maple@
mode=Maple@
allow2d=0@
name=Allow Cheating@
editing=useHTML@
```

In this example, the code verifies if the **factor** command has been used in the answer. The student response will be marked incorrect if the Maple **factor** command is used.

```
question=Factor(x^2-1)@
maple=evalb(0=StringTools[Search]("factor","$RESPONSE")) and evalb($RESPONSE=factor(x^2-1))@
type=maple@
mode=Maple@
allow2d=0@
name=Prevent Cheating@
editing=useHTML@
```

If you are concerned about students using Maple commands to calculate answers, consider using **Maple-graded Formula** to create your question. Students respond with a limited number of expressions and functions. As such, you do not need to explicitly check that the student response is a Maple command. For details, see the **Syntax** and **Student Response** subsections of *Maple-graded Formula (page 142)*.

See Also:

Maple-graded Question Type - Overview (page 140)

Maple Syntax (page 143)

Entering Chemistry Expressions

There are two ways to enter the answer to a question of the **chemistry** question type.

Text Mode

The system default mode is Text Mode, in which you enter symbolic or numeric math expressions in a simple keyboard notation, much like a graphing calculator.

Table 10.1: Guidelines for Entering Chemistry Expressions

Rule		Examples	
Superscripts and Subscripts	Enter superscripts using the caret "^" character, and subscripts using the underscore "_".	H_2O	H_2O
		H_2SO_4	H_2SO_4
Arrows in Equations	Use the text ->, <-, <=> for arrows.	->	→
		<-	←
		<=>	↔
Other Operators	Use "*" for the center dot operator.	*	·
	Use the "+" sign in equations.	+	+
	Note: *No other operations are allowed in equations. You can use the "-" sign to indicate ion charges (see below.)*		
Physical States, Ion charges, and Parentheses	Be sure to include physical states (in parentheses) if your equation requires them. Use the "+" and "-" characters for polarity and ion charges. Use parentheses to clarify interpretation of groups of characters.	Ca^2+(aq)+ (CO_3)^2- (aq) -> CaCO_3 (s)	
			$Ca^{2+}(aq) + CO^{2-}(aq) \rightarrow CaCO_3(s)$

Symbol Mode

For **chemistry** formula questions, you can also use Symbol Mode entry, which allows you to enter formulas in a WYSIWYG ("What you see is what you get") editing window. You can pick arrows, symbols or operators from a palette, and see your expressions appear as you type. Select **Equation Editor** in the Edit Question window of the Mathematical Formula question type to display the symbol palette.

Rules for Chemistry Expressions and Equations

- Spaces are ignored in expressions.
- Parentheses may be used to clarify physical states, compound groupings, and application of ion charges. Use the formula preview link to preview your expressions.
- In grading equations, any characters you enclose in parentheses (for example, ion charges, superscripts, subscripts, etc.) are associated with the preceding term (not the next term). Example: Pb^(2+) is Pb^{2+}
- For numbers that are not super or subscripted that precede a compound or element, the grader associates the number with the term that follows. Example: 2H_2O is $2H_2O$

See Also:

Chemistry (page 135)

Plotting a Student Response

Maple T.A. uses Maple plotting features to graph student responses or a function derived from a student response, for example, the definite integral of the student response. Maple provides many ways of representing data and mathematical expressions graphically using plots.

Plotting a student response is available only with the **Maple-graded Question Type**.

Plotting a Student Response

When entering code for plotting in the **Question Editor**, use only one Maple **plot** command. It must be the last command entered. You must use the long form name for Maple package functions.

Example

The following is the source file for a **Maple-graded Formula** question that plots the student response.

```
question=Enter an expression in x@
maple=true@
maple_answer=@
plot=plot($RESPONSE,x=0..10)@
type=formula@
mode=Maple@
name=2D Maple plot@
editing=useHTML@
```

The following is the source code for a **Maple Syntax** question that plots the student response using the Maple **tubeplot** command. Students must use Maple syntax in their response.

```
question=Enter a set of spacecurves using Maple syntax, for example, [cos(t),sin(t),0]@
maple=true@
maple_answer=@
plot=plots[tubeplot]($RESPONSE, t=0..4, radius=.05, numpoints=70)@
type=maple@
mode=Maple@
name=Tube Plot@
editing=useHTML@
```

Important:
You *must* use the long form name for Maple package functions. You must test your plot code prior to publishing assignments. Maple T.A. does not display error messages if the plot does not work correctly.

See Also:

Maple-graded Questions in the Question Editor (page 96)

Maple-graded Question Type - Overview (page 140)

Tutorial: Maple-based Algorithmic Question Authoring (page 244)

10.5 Using Formula Syntax

In the Maple T.A. system, you can create questions that use mathematical formulas or expressions. In the **Question Editor**, you can enter formulas using standard mathematical notation. In general, the system accepts the same formulas as a graphing calculator. Thus, the system accepts:

```
x^2-2x+1
2sin(x)
(x^2+1)e^(-x^2)
```

Note: The most common mistake is to forget parentheses "**( )**". For example, **1/(x+1)** is different from **1/x+1**, which the system interprets as **(1/x)+1**.

See Also:

Basic Math Syntax in the System (page 180)

Mathematical Functions & Operations (page 204)

10.6 Using the Equation Editor to Create Formatted Math Expressions

The equation editor, shown in **Figure 10.1**, provides an easy way to create formatted math expressions.

Editing with the Equation Editor

Using the Equation Editor, you can add symbolic math expressions and symbols to the question text and answer regions in all question types available through the Question Editor.

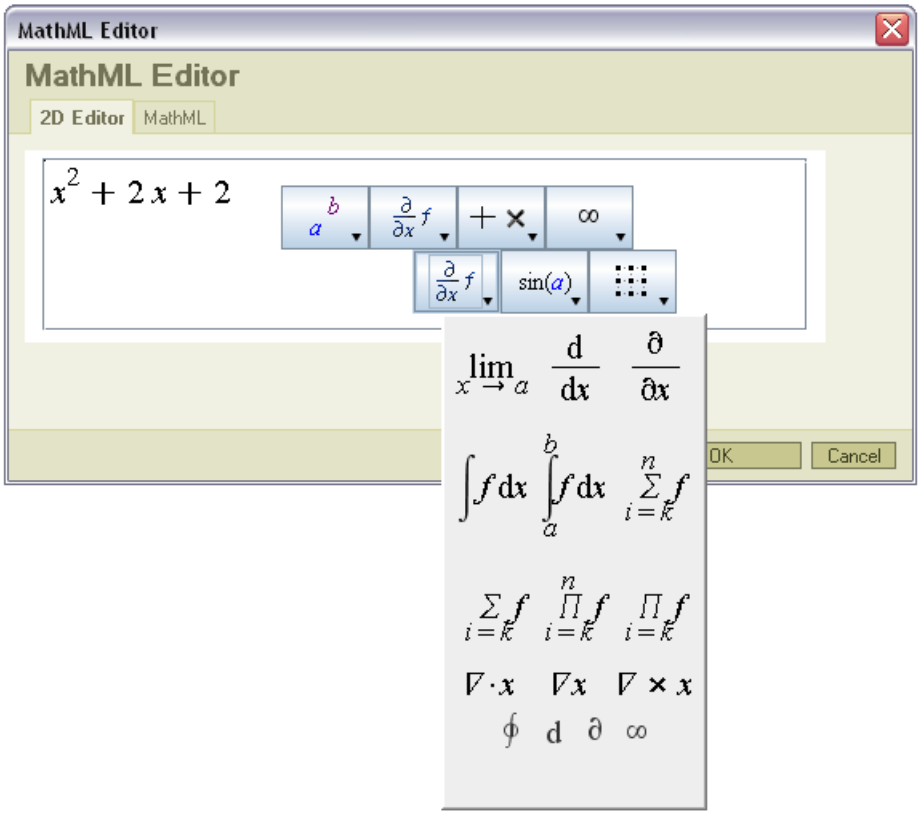


Figure 10.1: Equation Editor

Adding/Editing an Expression Using the Equation Editor

1. Click the question text region or the answer text region and select the Sigma icon, Σ, in the toolbar.
2. In the Equation Editor dialog, right-click (**Control**-click on Macintosh) the 2-D editor region to launch the Equation Editor palette. You can also add an expression using keyboard shortcuts.
3. Click **OK**.

Keyboard Shortcuts

The **Equation Editor** accepts keyboard shortcuts.

Table 10.2: Keyboard Shortcuts for the Equation Editor

Result	Shortcut Characters
Fraction	/
Superscript	^ (caret)
Subscript	_ (underscore key)
Pre-superscript	Ctrl+^
Pre-subscript	Ctrl+_

Template Shortcuts

A second set of keyboard shortcuts handles the insertion of templates:

- The **Tab** key cycles the cursor through the open template blanks in an expression.

- Enter the first few characters of an expression or symbol name and select **Ctrl** + **Space** (**Command** + **Space** on Macintosh) to insert the expression or symbol name, or to display a list of commands to complete the entry.

Equation Editor Palette

The Maple T.A. system uses a Equation Editor for creating and displaying symbolic characters and expressions. The **palette** question type enables you to use customized Math editor toolbars and palettes in creating student response objects.

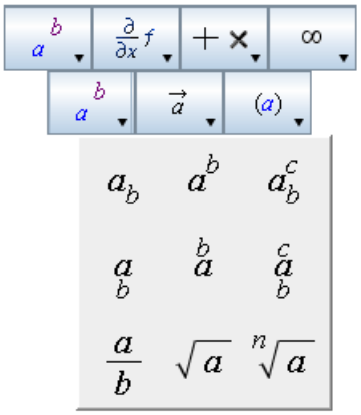
Using the Equation Editor Palette

The palette is used to insert equation templates and symbols. The standard palette contains one main panel organized into four categories: layout templates, expression templates, relational palettes, and symbol palettes. To display the palette, right-click (**Control**-click on Macintosh) the Equation Editor region (see **Figure 10.1**).

To display the options available in each panel, click the mouse over a template. A list of options displays. Continue to click the available palettes or scroll the mouse over the sub-templates. To insert a template, click the mouse over a highlighted template. The templates and character palettes are broadly organized by theme, such as *calculus templates* or *arrow symbols*. Spend a few minutes exploring the toolbar to familiarize yourself with the available templates and symbols.

Layout templates

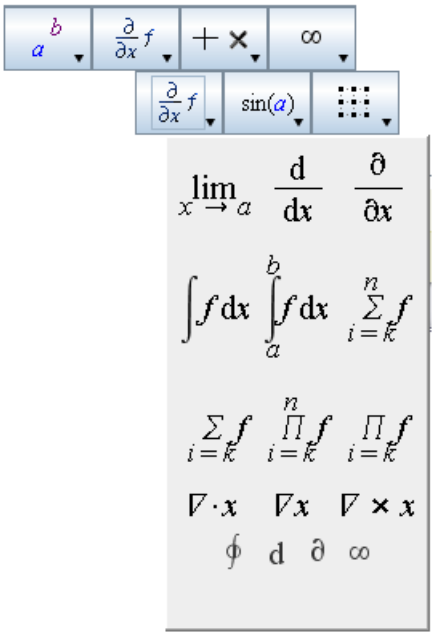
The left side of the toolbar consists of pop-up palettes containing an assortment of layout templates. Templates are highlighted as you move the mouse over them.



Layout Palette

Expression templates

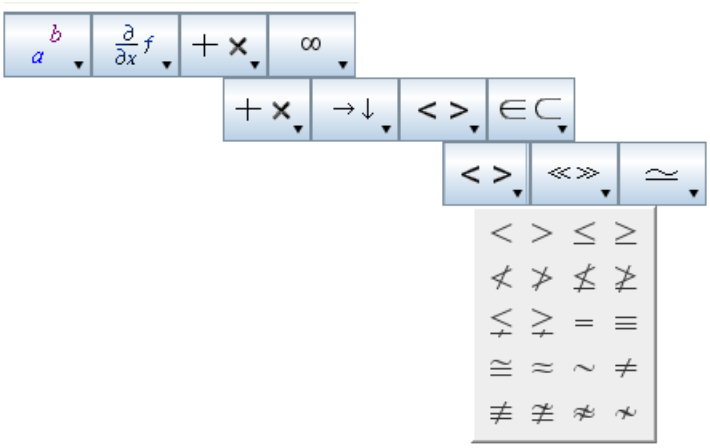
The second section of the standard toolbar consists of pop-up palettes containing expression templates.



Expression Palette

Relational palettes

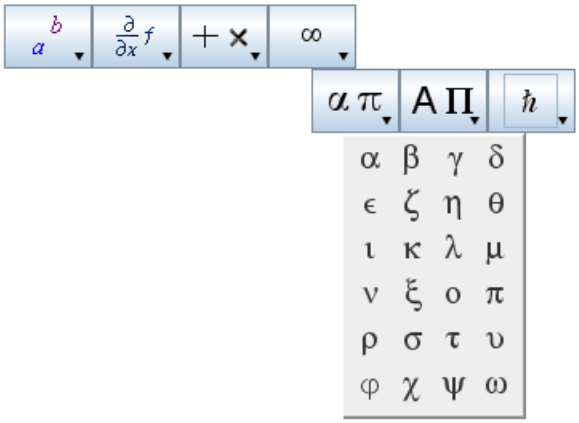
The third section of the standard toolbar consists of pop-up palettes containing relational templates.



Relational Palette

Symbol palettes

The fourth section of the standard toolbar consists of pop-up palettes containing symbol templates.



Symbol Palette

See Also:

Palette Questions (page 165)

Editing with the Equation Editor (page 188)

Equation Editor Requirements and Syntax

When creating an Equation Editor for Palette-based questions, below is a list of possible palette entries. Note: All entries are case-sensitive. If the palette item syntax is not correct, the text will display on the button instead of the proper symbol.

To create a new toolbar definition, you need to create a question bank in plain-text script format and add the toolbar definition at the beginning of the file using the **qu.env.palette.toolbarname=** structure (see below). You can define one or more toolbar definitions at a time. A toolbar definition can define one or more palette entries at a time.

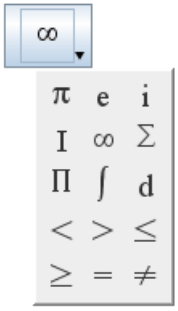
<btn> - name of the palette item.

<sub cols> - number of columns in the palette

<sep/> - palette separator

Table 10.3: Defining Palettes

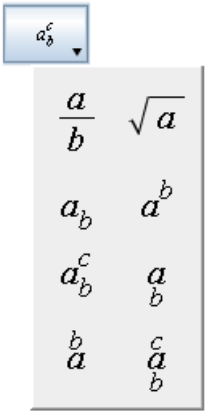
Palette Definition	Palette

`<btn>SYMBOLPALETTE</btn>`
`  <sub cols=3>`
`  <btn>pi</btn>`
`  <btn>e</btn>`
`  <btn>i</btn>`
`  <btn>I</btn>`
`  <btn>infin</btn>`
`  <btn>Sum</btn>`
`  <btn>Product</btn>`
`  <btn>int</btn>`
`  <btn>d</btn>`
`  <btn>lt</btn>`
`  <btn>gt</btn>`
`  <btn>le</btn>`
`  <btn>ge</btn>`
`  <btn>equals</btn>`
`  <btn>ne</btn>`
`  </sub>`
`<sep/>`
	

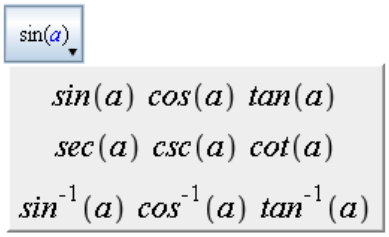
Palette Definition	Palette

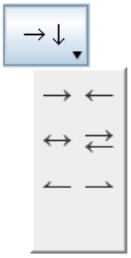
<btn>LCGREEKPALETTE</btn>
_{
<btn>alpha</btn>
<btn>beta</btn>
<btn>gamma</btn>
<btn>delta</btn>
<btn>lambda</btn>
<btn>mu</btn>
<btn>nu</btn>
<btn>psi</btn>
<btn>phi</btn>
<btn>sigma</btn>
<btn>Delta</btn>
<btn>Gamma</btn>
<btn>omega</btn>
<btn>epsilon</btn>
<btn>zeta</btn>
<btn>eta</btn>
<btn>theta</btn>
<btn>kappa</btn>
<btn>rho</btn>
<btn>upsilon</btn>
<btn>xi</btn>
}
<sep/>	

<btn>OPERATORPALETTE</btn>
_{
<btn>plus</btn>
<btn>minus</btn>
<btn>plusmn</btn>
<btn>times</btn>
<btn>middot</btn>
<btn>divide</btn>
}
<sep/>	

<btn>SUBSUP</btn>
_{
<btn>FRAC</btn>
<btn>SQRT</btn>
<btn>SUB</btn>
<btn>SUP</btn>
<btn>SUBSUP</btn>
<btn>UNDER</btn>
<btn>OVER</btn>
<btn>UNDEROVER</btn>
}
<sep/>
	

Palette Definition	Palette

<btn>TRIGPALETTE</btn>
_{
<btn>SIN</btn>
<btn>COS</btn>
<btn>TAN</btn>
<btn>SEC</btn>
<btn>CSC</btn>
<btn>COT</btn>
<btn>ARCSIN</btn>
<btn>ARCCOS</btn>
<btn>ARCTAN</btn>
}
	

<btn>ARROWPALETTE</btn>
_{
<btn>rarr</btn>
<btn>larr</btn>
<btn>harr</btn>
<btn>rlarr</btn>
<btn>lharu</btn>
<btn>rharu</btn>
}
	

Example Header Toolbar/Palette Definitions:

```
qu.env.palette.operator=
  <tb>
    <btn>OPERATORPALETTE</btn>
       <sub cols=1>
        <btn>times</btn>
        <btn>divide</btn>
       <btn>middot</btn>
      </sub>
  </tb>@

qu.env.palette.greek=
  <tb>
    <btn>SYMBOLPALETTE</btn>
  <sub cols=3>
    <btn>alpha</btn>
    <btn>beta</btn>
    <btn>gamma</btn>
  </sub>
  </tb>@
```

10.7 Using MathML to Create Formatted Math Expressions

Inline Display of Formatted Math Expressions

The algorithmic function **mathml(s)** accepts a string argument **s,** which it treats as a formula in calculator syntax and typesets it as MathML that can be used inside algorithms.

Note: When you create inline typeset math expressions using the **mathml(f)** function, the system automatically simplifies the expressions.

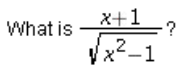
Example

```
qu.1.1.question=
  What is the derivative of ${mathml(sqrt(1+x^$n))}?
@
qu.1.1.answer=
  (1/2)(1+x^$n)^(-1/2)  $n x^${$n-1}
@
qu.1.1.algorithm=
  $n=range(2,10);
@
```

Since you can also use algorithmic commands to generate inline expressions, you can use the $mathml(s) command along with inline expressions of the form ${expression} to display the following typeset math expression inside a question statement.

```
What is ${mathml( (x+1)/sqrt(x^2-1) )}
```

This will generate



This is a quick and easy way to place MathML in question banks that use a little math notation but are not heavily reliant on it.

In addition, your expression can include other algorithmic functions and variables. You can use the following,

```
${mathml((x+$a)/sqrt(x^$n-1))}
```

Numeric Arguments to the mathml function

```
mathml(expr, 0)
```

returns MathML with no <math> tags

```
mathml(expr, 1)
```

(the default) returns MathML with <math> tags

See Also:

Simplified Math Expressions (page 199)

mathml(f) (page 226)

MathML Authoring with MathType

The Maple T.A. system can display and process MathML expressions as part of questions, answers, and feedback. However, creating MathML expressions in a plain text or HTML editor is extremely complicated, so this topic discusses how to use Design Science MathType™ Version 6 to create expressions that can then be exported into MathML expressions for importing into your Maple T.A. content.

Microsoft® Windows® and Mac OS® X users can use the Design Science MathType Version 6 to create MathML expressions. To do so, follow these steps:

1. Launch MathType.
2. Set the MathType MathML translator to export MathML as follows:
    a. From the menu, select **Preferences > Translators** to set MathML export.
    b. Set the options *exactly* as indicated in the **MathType** Dialog box below, then click **OK**:
        - Select **Translator: MathML 2.0 (no namespace)** radio button
        - Clear **Include translator name in translation** check box.
        - Clear **Include MathType data in translation** check box.

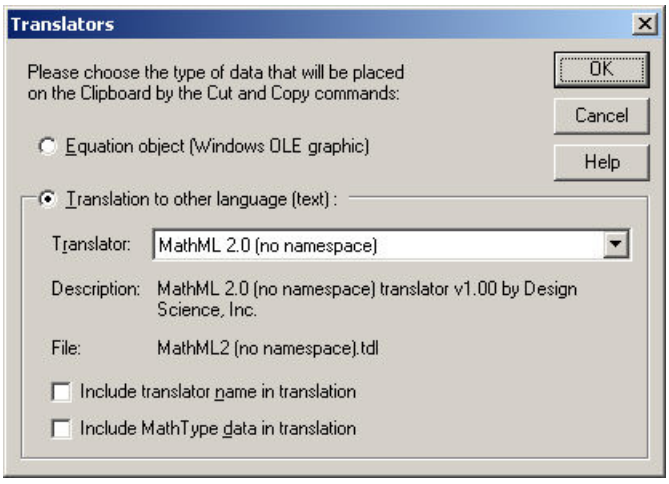

Translators
Please choose the type of data that will be placed on the Clipboard by the Cut and Copy commands:
OK
Cancel
Help
Equation object (Windows OLE graphic)
Translation to other language (text) :
Translator:
MathML 2.0 (no namespace)
Description:
MathML 2.0 (no namespace) translator v1.00 by Design Science, Inc.
File:
MathML2 (no namespace).tdl
Include translator name in translation
Include MathType data in translation

Figure 10.2: MathType Translator to MathML

3. Create your math expression using MathType.
4. Select the expression (use **Ctrl-A** to select all elements).
5. Copy the highlighted expression to the Windows clipboard (use **Ctrl-C**).
6. Open your **.qu** plain-text script file and position the cursor at the location for the MathML expression.
7. Paste the MathML expression into your plain-text script file (use **Ctrl-V**).
8. Save your **.qu** script file according to the system requirements.
9. Open your class web site and use the **Question Editor** to upload and install the new question bank, with MathML intact.

See Also:

Editing with the Equation Editor (page 188)

MathML in the Questions

MathML is used by the system for both the display of complicated math expressions as well as for *content* in math expressions (mathematical meaning).

For small expressions, you can use the **mathml(f)** function to create inline MathML expressions, in effect, to force the system to render an expression defined in calculator syntax as a typeset math expression.

Example

```
qu.1.8.mode=Formula@
qu.1.8.name=Fractions@
qu.1.8.editing=useHTML@
qu.1.8.algorithm=
$a=range(1,8);
$num=$mathml("(x/2)=$a");
$ans=$a*2;
@
qu.1.8.question=
Solve for x:
<br> $num@
qu.1.8.answer=$ans@
```

For larger expressions, you can include the MathML code for the expressions directly in questions, answers, comments, hints, and solutions in your question bank.

Example

The following is an example of MathML code used to create an expression inside a question script:

```
qu.1.5.mode=Equation@
qu.1.5.name=<b>Equations with algorithmic MathML feedback</b>@
qu.1.5.comment=
  Use the slope/intercept form of the line to find the equation as
  <math>
  <mo>(</mo><mi>y</mi>  <mo>-</mo>  <mn>$yone</mn><mo>)</mo>
  <mo>=</mo>
  <mfrac>
  <mrow><mn>$ytwo</mn>  <mo>-</mo>  <mn>$yone</mn></mrow>
  <mrow><mn>$xtwo</mn>  <mo>-</mo>  <mn>$xone</mn></mrow>
  </mfrac>
  <mo>(</mo><mi>x</mi>  <mo>-</mo>  <mn>$xone</mn><mo>)</mo>
  </math>
@
qu.1.5.editing=useHTML@
qu.1.5.algorithm=
  $xone=decimal(1, rand(0,10));
  $xtwo=decimal(1, rand(0,10));
  condition: ne($xone,  $xtwo);
  $yone=decimal(1, rand(0,10));
  $ytwo=decimal(1, rand(0,10));
  $m=($ytwo  - $yone)/($xtwo-$xone);
  $b=-$m*$xone  + $yone;
@
qu.1.5.question=

  What is the equation of the straight line passing through the points
  ($xone,  $yone) and  ($xtwo, $ytwo)?
<br>

@
qu.1.5.answer=
```

```
  y = $m x + $b
@
```

You can also generate MathML using Maple.

```
qu.1.5.mode=Equation@
qu.1.5.name=<b>Equations with algorithmic MathML feedback</b>@
qu.1.5.comment=
  Use the slope/intercept form of the line to find the equation as $Solution.
@
qu.1.5.editing=useHTML@
qu.1.5.algorithm=
  $xone=decimal(1, rand(0,10));
  $xtwo=decimal(1, rand(0,10));
  condition: ne($xone, $xtwo);
  $yone=decimal(1, rand(0,10));
  $ytwo=decimal(1, rand(0,10));
  $m=($ytwo - $yone)/($xtwo-$xone);
  $b=-$m*$xone + $yone;
$Solution=maple("printf(MathML:-ExportPresentation(y=$m*x + $b))");
@
qu.1.5.question=

  What is the equation of the straight line passing through the points
  ($xone, $yone) and ($xtwo, $ytwo)?
<br>

@
qu.1.5.answer=
  y = $m x + $b
@
```

See Also:

LaTeX Authoring Overview (page 247)

Inline Display of Formatted Math Expressions (page 196)

MathML Authoring with MathType (page 197)

Simplified Math Expressions

When you use the **mathml(f)** function to easily render typeset mathematics expressions, the system also engages a set of automatic simplification rules, as follows.

- Simple addition, subtraction, and multiplication are performed. For example, x+(2+3) are set as x+5 and 2*3*x as 6x.
- Additions and subtractions of zero are disregarded. For example, x+0 is x
- Addition of a negative number on the right becomes subtraction. For example, x+(-2) becomes x-2.
- Subtraction of a negative number on the right becomes addition. For example, x-(-2) becomes x+2.
- Multiplication by a number always comes first. For example, x*2 becomes 2x.
- Multiplication by 0 becomes 0. For example,0*x becomes 0.
- Multiplication by 1 and -1 handled correctly. For example, -1*x becomes -x.
- Division of two numbers simplified:
  - Denominator is always positive. For example, 3/-2 becomes -3/2.
  - Quotient of two integers is always put in lowest terms. For example, 3/6 becomes 1/2 and 6/3 becomes 2.

The best place to use these functions is in in-line expressions. For example, consider the algorithm:

```
$a=range(-10,10);
$b=range(-10,10);
$c=range(-10,10);
$d=range(-10,10);
condition: ne($a*$c, $b*$d);
```

You can set the question as follows.

```
Solve the simultaneous linear equations ${mathml("$a*x+$b*y")}=0
and ${mathml("$c*x+$d*y")}=0
```

Exceptions to Automatic Simplification

There may be situations in which the simplifications performed by **mathml(f)** may go beyond what you want. For example, you might want to typeset "2*3"4"x" literally. Automatic simplification is the system default.

To override the system defaults, use the optional numeric arguments for the **mathml(f)** function, which indicates that it is not to simplify math, as follows.

There are two pieces of **mathml(f)** behavior that can be controlled by a numeric argument:

- Whether to enclose the generated MathML in **<math>** tags - the default is to do so
- Whether to simplify math expressions - the default is to do so

Syntax

mathml(expr) - set expression in MathML using the defaults above

mathml(expr, k) - set expression in MathML using the options from the table below

Table 10.4: Options for the mathml Function

k	Include <math> tags	Simplify math
0 Default	yes	yes
1	no	yes
2	yes	no
3	no	no

Turning off the **<math>** tags allows you to generate a MathML snippet with **mathml(...)** that is to be used within a larger math layout that is generated either in raw MathML or from LaTeX. For example, in the LaTeX macros you can write:

```
$$ \int_1^\infty \var{mathml("$a+$b/x^2",  1)} dx $$
```

to display simplified algorithmic math inside an integral.

See Also:

Inline Display of Formatted Math Expressions (page 196)

mathml(f) (page 226)

10.8 Plotting Using Maple

You can use Maple plotting features to display plots. Maple provides many ways of representing data and mathematical expressions graphically using plots. **You can display a plot with any question type.**

- Displaying a Maple Plot with the plotmaple Command - Use the **plotmaple** command anywhere an algorithmic variable can be used, that is, in question text, hints, feedback, solution, and answer. For more information, see **Displaying a Maple Plot.**
- Plotting a Student Response - Use Maple plotting features to graph student responses or a function derived from a student response. For more information, see *Plotting a Student Response (page 187)*.
- Plotting Options for 2-D and 3-D Maple Plots - Many options are available when generating a 2-D or 3-D Maple plot. For more information, see *Setting Plot Options (page 235)*.

Displaying a Maple Plot

To display a plot within a question, use the **plotmaple** command.

1. From the **Question Editor/Select Question Type** screen, click the Algorithm **Add** button.
2. Enter the plotmaple algorithmic variable using the structure

```
$algorithmic_variable_name=plotmaple("plotstatement, libname='filename.lib', plotdevice='gif|jpeg',
 plotoptions='options'");
```

3. Click **Preview** or **Refresh** to preview the plot image. See **Figure 10.3**.
4. Click **OK** to return to the **Question Editor/Select Name and Type** screen.
5. Click **Next**.
6. In the text field for the question type, enter the text of the question along with the algorithmic variable name defined in step 2.

Note: The algorithmic variable can also be used in the feedback or hints section.

Using plot options, you can specify the height and width of the image without losing any quality in the image. For example,

```
$a=range(1,5);
$myplot=plotmaple("plot($a*sin(x),x=-Pi..Pi), plotoptions='height=250, width=250'");
```

With the plotmaple command, you can specify whether you want the image saved as a GIF or JPEG. Maple animations are animated GIFs, so you must use GIF for an animation. For example,

```
$myanimation=plotmaple("plots[animate](sin(a*x),x=-Pi..Pi,a=1..4), plotdevice='gif'");
```

You can use another algorithmic variable in the plot command. For example,

```
$myplot=plotmaple("plot(sin(x),x=-Pi..Pi), plotdevice='gif', plotoptions='height=250, width=250'");
```

Note: You must use the long form name for Maple package functions

You can see a plot immediately using the Algorithm Editor.

```
$a=range(1,5);
$myplot=plotmaple("plot($a*sin(x), x=-Pi..Pi),plotdevice='gif',
plotoptions='height=250,width=250'");
```

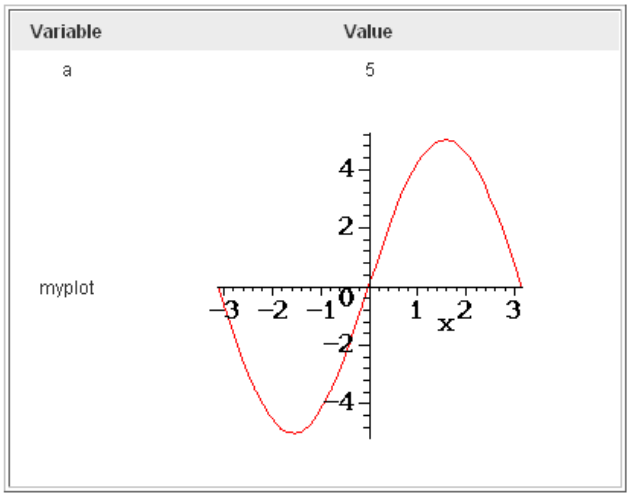


Figure 10.3: Plot with Algorithmic Variables

Example Using An Essay Question Type

The following is the source file for an **Essay** question type that displays a plot.

```
mode=Essay@
editing=useHTML@
algorithm=$wave=plotmaple("plot(sin(x),x=0..10)");@
question=Discuss the following wave.<br />
$wave@
```

Example Using A Randomly Generated Variable

The following is the source file for a **Maple-graded Formula** question type that displays a plot.

```
question=Determine the expression in the following plot.<br  />
$graph@
maple=evalb(($ANSWER)-($RESPONSE)=0);@
maple_answer=x^$n@
type=formula@
mode=Maple@
name=Displaying a Plot@
editing=useHTML@
algorithm=$n=maple("randomize():rand(1..10)()");
$graph=plotmaple("plot(x^$n,x=-2..2)");@
```

Example Using the Maple tubeplot Command

The following is the source file for a True-False question type that displays a tube plot.

```
mode=True False@
name=tube plot@
editing=useHTML@
algorithm=$algorithmic_variable_name=plotmaple("plots[tubeplot](  [t*cos(3*Pi*t), t*sin(3*Pi*t),
 t], t=0..4, radius=.05, numpoints=70)");@
question=<p class="Listing">This displays a tube plot using the Maple long form name for the
plots package.</p>
<p>$algorithmic_variable_name</p>@
answer=1@
choice.1=True@
choice.2=False@
fixed=@
```

Important: You must test your plot code prior to publishing assignments. Maple T.A. does not display error messages if the plot does not work correctly.

The drawMaplePlot command

Important: The Maple **drawMaplePlot** command has been superseded by the **plotmaple** command. Questions created using the **drawMaplePlot** command will continue to function. However, it is recommended that you use the **plotmaple** command.

To display a plot with a question using the drawMaplePlot command:

1. From the **Question Editor/Select Name and Type** screen, select the **Use HTML formatting for the question text** option.
2. Click **Next**.
3. In the text field for the particular question type, click the **Source** button and enter the HTML script tags and structure illustrated below.

```
<script>drawMaplePlot("maplecode",width,height,"libname");</script>

<script>drawMaplePlot("plot(sin(x), x=0..10)",400,400);</script>
```

The script requires the following syntax:

- <script> and </script> tags
- **drawMaplePlot** command
- Expression enclosed parentheses
- Maple code to create plot, enclosed in quotes
- Closing semicolon
- **width** (optional) - specifies plot width
- **height** (optional) - specifies plot height
- **libname** (optional) - specifies path to Maple libraries

To use multiple libraries, place all libraries in the same directory and specify the path using the **libname** option.

See Also:

plotmaple(text) (page 227)

Maple-graded Questions in the Question Editor (page 96)

Maple-graded Question Type - Overview (page 140)

Tutorial: Maple-based Algorithmic Question Authoring (page 244)

10.9 Mathematical Functions & Operations

The following tables provide a reference of mathematical functions, constants, and operators.

- **Table 10.5** contains general functions and numbers that can be used for both Maple T.A. and Maple questions.
- **Table 10.6** provides the function names for Maple T.A. and Maple questions for the hyperbolic and inverse hyperbolic functions. Note that Maple T.A. and Maple use different notation for these functions.
- **Table 10.7** contains exclusive Maple notation. For a complete list of Maple functions, refer to Maple documentation. Some Maple commands take unevaluated forms. For a discussion on student use of unevaluated forms, see *Maple Syntax (page 143)*.

Table 10.5: Mathematical Functions and Operators

Arithmetic		Trigonometric Functions	
`+`	addition	`sin`	sine
`-`	subtraction	`cos`	cosine
`*`	multiplication	`tan`	tangent
`/`	division	`arcsin`	inverse sine
`^`	exponentiation	`arccos`	inverse cosine
Numbers and Constants		`arctan`	inverse tangent
e	2.71828...	`sec`	secant
`pi (π)`	3.14159...	`csc`	cosecant
`2.9E8`	scientific notation: 290,000,000	`cot`	cotangent
Functions		`arcsec`	inverse secant
`sqrt`	square root	`arccsc`	inverse cosecant
`log`	logarithm base 10	`arccot`	inverse cotangent
`ln`	natural logarithm		
`abs`	absolute value function		

Table 10.6: Hyperbolic and Inverse Hyperbolic Functions

Function	Maple T.A. Notation	Maple Notation	Function	Maple T.A. Notation	Maple Notation
hyperbolic sine	`hypsin`	`sinh`	inverse hyperbolic sine	`archypsin`	`arcsinh`
hyperbolic cosine	`hypcos`	`cosh`	inverse hyperbolic cosine	`archypcos`	`arccosh`
hyperbolic tangent	`hyptan`	`tanh`	inverse hyperbolic tangent	`archyptan`	`arctanh`
hyperbolic secant	`hypsec`	`sech`	inverse hyperbolic secant	`archypsec`	`arcsech`
hyperbolic cosecant	`hypcsc`	`csch`	inverse hyperbolic cosecant	`archypcsc`	`archcsch`
hyperbolic cotangent	`hypcot`	`coth`	inverse hyperbolic cotangent	`archypcot`	`arccoth`

Table 10.7: Maple Notation

Constant	Maple Notation
2.71828...	`exp(1)`
3.14159...	`Pi`

$\sqrt{-1}$, the square root of -1	`I`
Function	**Maple Notation**
logarithm base 10	`log10`
integral	`Int`
differential equation	`Diff`
limit	`Limit`
exponential function Calculates the value of **e** to power of x, where **e** is the base of the natural logarithm, 2.718281828...	`exp`

Note: The variable **e** should *not* be assigned as a global variable in **Maple** code. It can be used, however, as a local variable.

See Also:

Maple Syntax (page 143)

10.10 Using a Table of Units

Default Table of Physical Unit Equivalents

The system recognizes many, but not all, units. To use equivalent units that are not recognized, create a custom table of units for your question bank.

The following tables give the default units recognized by the system.

Unit	Definition	Name
Base Units		
m		meter
s		second
kg		kilogram
A		amp
K		kelvin

Derived Units		
ng	10^-9 g	nanogram
ug	10^-6 g	microgram
mg	10^-3 g	milligram
g	0.001kg	gram
Mg	10^6 g	megagram
lb	453.59237 g	pounds (Avoirdupois)
lbs	lb	pounds (Avoirdupois)
oz	(1/16)lb	ounces (Avoirdupois)
deg	Pi/180	degree
degree	Pi/180	degree
degrees	Pi/180	degree
rad	1	radian
radian	1	radian
radians	1	radian
nm	10^-9 m	nanometer
um	10^-6 m	micrometer
mm	10^-3 m	millimeter
km	10^3 m	kilometer
Mm	10^6 m	megameter
meter	m	meter
meters	m	meter
cm	0.01 m	centimeter
centimeter	cm	centimeter
centimeters	cm	centimeter
ft	m*0.3048	foot
foot	ft	foot

feet	ft	foot
in	(1/12)ft	inch
inch	in	inch
inches	in	inch
yd	3feet	yard
yard	yd	yard
yards	yd	yard
mi	5280 feet	miles (statute)
mile	mi	miles (statute)
miles	mi	miles (statute)
L	m^3/1000	liter
nL	10^-9 L	nanoliter
uL	10^-6 L	microliter
mL	10^-3 L	milliliter
kL	10^3 L	kiloliter
ML	10^6 L	megaliter
liter	L	liter
liters	L	liter
sec	s	second
second	s	second
seconds	s	second
min	60s	minute
minute	min	minute
minutes	min	minute
h	60min	hour
hr	h	hour
hour	h	hour
hours	h	hour

ns	10^-9 s	nanosecond
us	10^-6 s	microsecond
ms	10^-3 s	millisecond
ks	10^3 s	kilosecond
Ms	10^6 s	megasecond
N	kg m/s^2	newton
nN	10^-9 N	nanonewton
uN	10^-6 N	micronewton
mN	10^-3 N	millinewton
kN	10^3 N	kilonewton
MN	10^6 N	meganewton
J	N m	joule
nJ	10^-9 J	nanojoule
uJ	10^-6 J	microjoule
mJ	10^-3 J	millijoule
kJ	10^3 J	kilojoule
MJ	10^6 J	megajoule
Pa	N/m^2	pascal
nPa	10^-9 Pa	nanopascal
uPa	10^-6 Pa	micropascal
mPa	10^-3 Pa	millipascal
kPa	10^3 Pa	kilopascal
MPa	10^6 Pa	megapascal
W	J/s	watt
nW	10^-9 W	nanowatt
uW	10^-6 W	microwatt
mW	10^-3 W	milliwatt

kW	10^3 W	kilowatt
MW	10^6 W	megawatt
Hz	1/s	hertz
nHz	10^-9 Hz	nanohertz
uHz	10^-6 Hz	microhertz
mHz	10^-3 Hz	millihertz
kHz	10^3 Hz	kilohertz
MHz	10^6 Hz	megahertz
C	A s	coulomb
nC	10^-9 C	nanocoulomb
uC	10^-6 C	microcoulomb
mC	10^-3 C	millicoulomb
kC	10^3 C	kilocoulomb
MC	10^6 C	megacoulomb
nA	10^-9 A	nanoamp
uA	10^-6 A	microamp
mA	10^-3 A	milliamp
kA	10^3 A	kiloamp
MA	10^6 A	megaamp
V	J/C	volt
nV	10^-9 V	nanovolt
uV	10^-6 V	microvolt
mV	10^-3 V	millivolt
kV	10^3 V	kilovolt
MV	10^6 V	megavolt
ohm	V/A	ohm
nohm	10^-9 ohm	nanoohm

nohm	10^-9 ohm	nanoohm
uohm	10^-6 ohm	microohm
mohm	10^-3 ohm	milliohm
kohm	10^3 ohm	kiloohm
Mohm	10^6 ohm	megaohm
ohms	ohm	ohm
S	A/V	siemens
nS	10^-9 S	nanosiemens
uS	10^-6 S	microsiemens
mS	10^-3 S	millisiemens
kS	10^3 S	kilosiemens
MS	10^6 S	megasiemens
F	C/V	farad
nF	10^-9 F	nanofarad
uF	10^-6 F	microfarad
mF	10^-3 F	millifarad
kF	10^3 F	kilofarad
MF	10^6 F	megafarad
Wb	V s	weber
nWb	10^-9 Wb	nanoweber
uWb	10^-6 Wb	microweber
mWb	10^-3 Wb	milliweber
kWb	10^3 Wb	kiloweber
MWb	10^6 Wb	megaweber
T	Wb/m^2	tesla
nT	10^-9 T	nanotesla
uT	10^-6 T	microtesla

H	Wb/A	henry
nH	10^-9 H	nanohenry
uH	10^-6 H	microhenry
mH	10^-3 H	millihenry
kH	10^3 H	kilohenry
MH	10^6 H	megahenry
lm	cd	lumen
lx	cd/m^2	lux
mol	6.02214199 * 10^23	mole
mole	6.02214199 * 10^23	mole
moles	6.02214199 * 10^23	mole
M	mol/L	molar
Molar	mol/L	molar

Other Units		
degC	(T + 273.15) K	degree Celsius
degF	(5/9)(T - 32) + 273.15 K	degree Fahrenheit

Custom Table of Equivalent Units

You can specify a table of custom equivalent units, which you can use in any Numeric question in the question bank. You can define new units, or add additional names or symbols for base units.

Defining a Table of Equivalent Units for a Question Bank

To define a table of units for a question bank, you need to create a question bank in plain text format and include an equivalent units table at the beginning of file. This table requires the following structure:

```
qu.env.numeric.<tablename>=
  <table_definition>
@
```

where <**tablename**> is a string of lower-case letters (without spaces) identifying the table. Once the table of units is defined, you need to upload the question bank.

Table Structure

A table of unit definitions **<table_definition>** has the general structure:

```
<new_unit1>  = <numeric_multiplier1><old_unit1>, description1;
<new_unit2>  = <numeric_multiplier2><old_unit2>, description2;
```

For example:

```
gr  = 1m, grain (metric);
Gr  = 0.00006479891kg, grain (troy);
```

Note: All unit definitions must reduce to the base SI units: **m** (meter), **kg** (kilogram), **s** (second), **A** (amp), **K** (kelvin), and **cd** (candela). Units can be derived in terms of other units, for example, **micron = 0.0001cm**, provided that you include the line **cm = 0.01m** (or otherwise relate **cm** with the base SI unit **m**) in your table.

Include Table-specific Help

Optionally, there can be a line

```
qu.env.numeric.tablename.help=http://myhelpurl@
```

where **myhelpurl** is the URL of a page with help information on the units in the table.

Using a Custom Table of Unit Equivalents in a Question

To use a table of units, include the following statement in the question definition of any Numeric question in your question bank.

```
qu.<topic_number>.<question_number>.units=<tablename>@
```

Replacing Derived Units with Custom Units in Units Table

When custom units are used in a question, the custom units are automatically appended to the table of base and derived units available to students.

To instead replace the derived units in this table with the custom units (base units are always displayed), include the following statement in the question definition.

```
qu.<topic_number>.<question_number>.appendUnits=false@
```

See Also:

Using a Table of Units (page 205)

Numeric (page 146)

10.11 Tutorial: Displaying Answers vs. Formulas for Answers

After grading, for questions answered incorrectly, you can display the:

- *calculated value* of the answer field
- *formula* used to define the answer

Depending on your instructional objectives, you can display one or both.

Displaying the answer Field Formula:

In some cases, you may feel that there is extra instructional benefit to be derived from showing the student the correct formula instead of the final answer. If you do not use the *comment* field in your question definition, the default system behavior after grading is to display the value of the *answer* field. To return the formula, assign it to the *answer* field. If the *answer* field is defined by a formula using variables, the system displays a formula, with algorithmically generated data substituted for the variables.

Example 1 If a student gives the wrong answer to the question:

```
qu.1.topic=Displaying Answers vs Formulas for Answers@
qu.1.1.mode=Formula@
qu.1.1.name=Kinetic energy of random object@
qu.1.1.algorithm=
  $mass=decimal(1,rand(1,  10));
  $velocity=decimal(1,rand(10,  15));
  @
qu.1.1.question=
An object of mass $mass kg is moving at a speed of $velocity m/s. What is its kinetic energy
(in joules)? @
qu.1.1.answer= (1/2)$mass*($velocity)^2@
```

the system displays the correct answer as:

Correct answer: (1/2) 5.4 * (11.3)^2

(assuming that the randomization set **$mass=5.4** and **$velocity=11.3**).

Displaying the Calculated Final Answer:

In some cases, you will prefer to return only the final answer.

To return only the final answer:

1. Do not use the *comment* field in your question definition.
2. Create a variable, **$ans**, that holds the formula for the answer.
3. Use the **$ans** variable in the *answer* field instead of the formula for the answer.

Example 2 If a student gives the wrong answer to the question:

```
qu.1.2.mode=Formula@
qu.1.2.name=Show only final calculated answer@
qu.1.2.editing=useHTML@
qu.1.2.algorithm=
$mass=rand(4,8,2);
$velocity=rand(10,20,2);
    $ans=sig(2,  (1/2)$mass * ($velocity)^2)
@
```

```
qu.1.2.question=
What is the kinetic energy of a body of mass $mass kg moving at $velocity m/s?@
qu.1.2.answer=$ans@
```

the system displays the correct answer as:

Correct answer: 900

(assuming that the algorithmically generated variables set **$mass=4.5** and **$velocity=20**).

In this case an incorrect response by the student shows only the correct numerical answer (to 2 significant figures).

Displaying Both the Formula for the Answer and the Final, Calculated Answer:

To show the student both the correct formula and the calculated answer, combine these methods and use the *comment* field.

1. Create a variable, **$ans**, that holds the formula for the answer.
2. Use the **$ans** variable in the *comment* field, along with the formula for the answer.

Example 3 If a student gives the wrong answer to the question:

```
qu.1.3.mode=Formula@
qu.1.3.name=Shows both calculated and formula answers@
qu.1.3.comment=
<br>This answer is derived according to the general formula,
<br>kinetic energy = (1/2)mass * (velocity<sup>2</sup>)
<br>In this case, the kinetic energy is
<br>(1/2)$mass * $velocity<sup>2</sup>, or $ans.
@
qu.1.3.editing=useHTML@
qu.1.3.algorithm=
$mass=rand(4,8,2);
$velocity=rand(10,20,2);
    $ans=sig(2,  (1/2)$mass * ($velocity)^2)
@
qu.1.3.question=
What is the kinetic energy of a body of mass $mass kg moving at $velocity m/s?@
qu.1.3.answer=$ans@
```

the system displays the correct answer as:

Correct answer:

This answer is derived according to the general formula,

kinetic energy = (1/2)mass * (velocity^2)

In this case, the kinetic energy is

(1/2)7.6 * 19^2, or 1,400.

(assuming that the algorithmically generated variables set **$mass=7.6** and **$velocity=19**).

Consider another example.

Example 4

```
qu.1.4.mode=formula@
qu.1.4.algorithm=
    $a=rint(12)+1;
$b=rint(12)+1;
```

```
$c=rint(12)+1;
@
qu.1.4.question=
Find the solution of the linear equation
${a}x + ${b}=${c}.
@
qu.1.4.answer=(${c}-${b})/${a}@
```

The above code chooses random integer coefficients **$a**, **$b**, and **$c** between 1 and 12. The formula for the answer is calculated in the *answer* field.

Note: Braces **{variable}** is used to enclose the variable names in expressions. This structure allows you to use variable names directly adjacent to other letter characters.

Although the above example does not cause errors, there are several side effects:

- The answer is displayed in the form **(7-3)/5** (for example). This provides an indication of the method for obtaining the answer, which may be inappropriate.
- The above format can return answers of the form **(7-7)/5**, which is more complicated than the simplified form, **0**.

To avoid these issues, rewrite this example as follows, using a new variable **$ans** (so that you return only the final answer (to 3 decimal places)):

Example 5

```
qu.1.5.mode=formula@
qu.1.5.algorithm=
$a=rint(12)+1;
$b=rint(12)+1;
$c=rint(12)+1;
$ans=decimal(3,($c-$b)/$a);
@
qu.1.5.question=
     Find the solution of the linear equation
${a}x+${b}=${c}.
@
qu.1.5.answer=$ans@
```

See Also:

Adding and Editing Comments (page 78)

10.12 Tutorial: Formatting Integers and Reals

This tutorial demonstrates the effect of applying various formatting options to the same expression in a question.

Consider a question that contains the text:

A dog weighs **$m** kg.

The following table shows how different assignments of values to **$m** are interpreted.

Table 10.8: Formatting Numbers with $m

Syntax	Format Display
`$m=15.27375491`	A dog weighs 15.27375491 kg.
`$m=sig(3, 15.27375491)`	A dog weighs 15.3 kg.

Syntax	Format Display
`$m=int(15.27375491)`	A dog weighs 15 kg.
`$m=decimal(1, 15.27375491)`	A dog weighs 15.3 kg.
`$m=decimal(0, 15.27375491)`	A dog weighs 15 kg.

```
decimal(n, x)
```

Returns **x** expressed as a floating-point number rounded to **n** decimal places.

```
sig(n, x)
```

Returns **x** expressed as a floating-point number rounded to **n** significant digits.

```
int(x)
```

Returns the integer part of **x**.

10.13 Tutorial: Setting Margin of Tolerance in Non-numeric Questions

Note: Use the **Question Editor** on-screen dialog boxes for setting the margin of error in numeric and numeric-with-unit questions. For more information, see *Setting Answer Format in Numeric Questions (page 85)*.

For an introduction to the concept of *margin of error*, see *Controlling Answer Tolerance (page 183)*.

For basic information on the system plus or minus operator, "**?**", see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

Absolute Errors

You can specify an absolute error using the **?** operator.

x ? y represents **x +/- y**.

Example 1

```
2.12 ? 0.01
```

specifies a margin of error of +/- **0.01** in the answer. The answer matches any response between **2.11** and **2.13** (inclusive). For example, the system grades **2.114** or **2.123456789**, as well as **2.11** and **2.13**, correct.

Example 2

```
2.1J ? 0.25J
```

specifies a margin of error of +/- **0.25J**. The system grades correct any response between **1.85J** and **2.35J**. The error must have the same dimension as the correct answer.

Example 3

```
cos(pi/6) ? 1000
```

specifies a margin of error of +/- **1000**. The system grades correct any response between **cos(pi/6)-1000** and **cos(pi/6)+1000**, approximately between **-999.1339746** and **1000.866025**.

Percentage Errors

You can also specify a percentage error using the **?** operator.

x (1 ? y) represents **x +/- x*y**.

Example 4

```
1.52 (1 ? 0.025)
```

specifies a margin of error of **2.5%**. The system grades correct any response between **1.444** and **1.596**.

Example 5

```
50 (1 ? 1.1)
```

specifies a margin of error of **110%**. The system grades correct any response between **-5** and **105**.

Algorithmic Variables

You can use a margin of error with answers that are algorithmic variables. Consider a question in which the correct answer is assigned to **$answer**.

Note: The **?** operator cannot be used in an algorithmic variable definition (or other algorithmic statement).

Example 6

To specify an absolute margin of error of **0.1**, use:

```
$answer ? 0.1
```

Example 7

To specify a percentage margin of error of **3%**, use:

```
$answer (1 ? 0.03)
```

Using the ? Operator in Complex Expressions

You can use the **?** operator in complex expressions. It is an arithmetic operator.

Example 8

```
2.12?0.01
```

matches any response between **2.11** and **2.13** and

```
2*(2.12?0.01)
```

matches any response between **4.22** and **4.26**.

Example 9

The ordinary rules of arithmetic (for example, the distributive rule) apply.

```
2*(2.12?0.01)
```

is equivalent to:

```
(2*2.12)?(2*0.01)
```

Example 10

You can specify very complicated ranges of numbers.

```
((2.17 ? 0.01) + e^(1.07?2.03))/(1? 0.5)
```

Important: If the exact answer is an expression, not a simple formula, number, or variable, ensure that you use parentheses to generate the correct answer.

Example 11

```
2 + 2 * (1 ? 0.05)
```

which defines a tolerance of **5%** of **2**, is different from:

```
(2 + 2) * (1 ? 0.05)
```

which defines a tolerance of **5%** of the sum **4**.

See Also:

Answer Tolerance Example Script (page 285)

Controlling Answer Tolerance (page 183)

10.14 Overview of Algorithmic Question Capabilities

Algorithmic question generation allows you to write questions in which specified variables are assigned random values (according to defined ranges and conditions) each time the question is reused in an assignment. Using algorithmically generated variables within questions allows you to generate multiple question permutations from the same question *stem*. Thus, you can write a single *stem* or *template* question that can be reused. The numbers and wording change to use new randomly determined values in different assignment attempts (by the same student or different students).

The system allows you to create variables that use mathematical/logical arguments to generate random numbers in questions (where the data used in a question is determined by random selection from an entire scenario's data and the variables are based on lists), or to use a combination of mathematical/logical and list-based variables to produce a randomized question.

See Also:

Rules for Naming Variables (page 221)

Tutorial: Algorithmic Question (page 241)

Tutorial: Maple-based Algorithmic Question Authoring (page 244)

Tutorial: List-based Variables (page 242)

10.15 How Does an Algorithmic Question Work in an Assignment?

Authoring

1. Create a static question -- a version of the question in which the variables have fixed values.
2. Edit the question. (All three authoring methods support algorithm authoring.)
3. Click **Edit** in the **Algorithmic Questions** region.
4. Add variable(s) and click **Save**.
5. Click **Next** and replace the fixed values with variable names in the question. Variable names must start with the **$** (dollar sign) character.
6. Save your question bank.

Assignment Creation

1. Select algorithmic questions for inclusion in an assignment in the same manner used for other questions.
2. In the **Assignment Editor**, each time you preview the question the system regenerates the variable values. To preview the question with new variable values, preview it again.
3. Save your assignment.

Assignment Use

1. Each time the assignment is instantiated (generated for a new session) by Maple T.A., new variable values are created in all the algorithmic questions. New assignment sessions occur either when the same student is making a new attempt at the same assignment or when different students request sessions from the same assignment.
2. Until an assignment is submitted for grading, variable data that is created for that particular session persists throughout the assignment session, even when students navigate between questions.
3. To give individual students the ability to return and rework an assignment with the same algorithmically generated variable values for each repeated attempt, use the **Reworkable** option for homework in Step 3 of the **Assignment Editor**.
4. Algorithmically generated variable values are recorded permanently in the graded session results stored in the class Gradebook. At any time following grading, students and instructors can return and review the exact assignment, with the variable values used and student responses entered in the assignment.

10.16 Working with Variable Data

For each type of question supported, you can include algorithmically generated (random) variables in question statements, answers, hints, comments, and solutions. Using variables in questions, you can generate variations on a single *template* question.

Types of Variables

- **Maple™-based variables:** You can use *any* Maple function to generate variables. See the tutorial on this topic, *Tutorial: Maple-based Algorithmic Question Authoring (page 244)*.
- **Range-based variables:** You can specify mathematical ranges (set by declaring minimum and maximum data values and increments for randomly generated data). Range-based variables are defined by a formula, and can include mathematical operations. For example, you can specify a variable **x** that is a random integer between

0 and 30, in steps of 3, using the variable definition: **$x=range(0,30,3)**. See the tutorial on this topic, *Tutorial: Algorithmic Question (page 241)*.

- **List-based variables:** You can randomly select a set of data from lists, and present to students a scenario with related variables. For example, you can create an astronomy question that randomly selects a planet in our solar system, and then automatically assigns the acceleration due to gravity at the planet's surface to the answer variable. See the tutorial on this topic, *Tutorial: List-based Variables (page 242)*.

You can use more than one type of variable in a question. You can use any combination.

You can place conditions on variables. For example, you can specify that the value of a variable not exceed that of another.

Note: You can create or edit variables and algorithmically generated data using the Algorithm Designer in the **Question Editor**. Alternatively, you can create questions without variable data using the **Question Editor**, and then export the plain-text script file (with the file extension **.qu**) to your hard drive. Add variables by editing the questions directly in your text or HTML editor.

See Also:

Generating Random Numbers in Questions (page 221)

Mathematical and Logical Operators in Algorithms and Answers (page 233)

The Algorithm Designer (page 80)

Using Script Files to Author Questions (page 283)

Functions within Algorithms

Algorithmic definitions of variables are of the form:

```
qu.x.y.algorithm=
$variable1=expression;
$variable2=expression;
@
```

- The use of the semicolon at the end of the variable definition is optional if there is only one variable defined in the algorithm statement.
- The syntax of formulas and the use of operators in variable definition statements is similar to the standard graphing calculator syntax used in the system, with the addition of some new functions. Therefore, the system's usual arithmetic operations and functions can be used in the **expression** portion of the code statement.
- However, unlike most other programming languages, variables cannot be redefined in terms of themselves (for example, the statement **$a=$a+1;** is invalid).

The following additional functions and operations are also available for use in variable definitions in algorithms.

Random Number Functions

Random integer generation (integers in a specified range, in steps of **k**):

- *Random Integer Generation rint(n), rint(m, n), rint(m, n, k) (page 230)*
- *Random Integer Generation range(n), range(m, n), range(m, n, k) (page 229)*

Random real number generation (to **k** significant digits):

- *rand(m, n), rand(m, n, k) (page 228)*

Operations and Modifiers

condition:x	See *condition:x (page 222)*

eq(a, b) ne(a,b)	See *eq(a, b), ge(a, b), le(a, b), ne(a, b) (page 223)*
not(a)	See *not(a) (page 227)*
gt(a, b) lt(a, b)	See *gt(a, b), lt(a, b) (page 225)*
if(a, b, c)	See *if(a, b, c) (page 225)*
decimal(n, x) int(x) sig(n, x)	See *decimal(n, x), sig(n, x), int(x) (page 223)*
lsu(n, x)	See *lsu(n, x) (page 226)*
sum(varname, start, stop, expr)	See *sum(var name, start, stop, expr) (page 232)*
switch(n, a, b, c, ...)	See *switch(n, a, b, c, ...) (page 232)*

See Also:

Rules for Naming Variables (page 221)

Inline Variable Expressions

You can include a computed expression anywhere within a question statement, comment, or hint.

- Inline variable expressions can contain computations involving functions or algorithmic variables defined elsewhere in the question.
- The syntax is to enclose the expression in **${...}**.

For example, you can include expressions like **${$n + 1}** in place of defining a separate variable to hold **n+1**. This simplifies complex algorithms.

Example

```
qu.1.1.question=
  What is the derivative of x^$n?
@
qu.1.1.answer=
  $n x^${$n-1}
@
qu.1.1.algorithm=
  $n=range(2,10);
@
```

See Also:

mathml(f) (page 226)

Inline Display of Formatted Math Expressions (page 196)

Simplified Math Expressions (page 199)

Rules for Naming Variables

Follow these rules when naming variables:

- A variable name must be of the form **$name** or **${name}**.
- The string **name** can consist of letters (**a**-**z** and **A**-**Z**), numbers (**0**-**9**), and underscore characters (_).
- The first character of **name** must be an alphabetic character (**a**-**z** or **A**-**Z**).
- If the **${name}** form is used, spaces can also appear in **name**. This allows you to use multiple word names without the underscore character.

Note: The variable **e** should *not* be assigned as a global variable in **Maple** code. It can be used, however, as a local variable.

See Also:

Functions within Algorithms (page 219)

10.17 Generating Random Numbers in Questions

The algorithm syntax in the system allows you to generate random numbers in a variety of ways, depending on your requirements and preferences. This table summarizes functions related to random number generation.

Table 10.9: Random Number Generators

Function	Range of Operation	Example	Description
`rint(n)`	0, ..., n-1	**rint(3)** = 0, 1, or 2	Returns a random integer between 0 and n-1 (inclusive). Generates n variations.
`rint(m,n)`	m, ..., n-1	`rint(3,6)` = 3, 4, or 5	Returns a random integer between m and n-1 (inclusive). Generates n-m variations.
`rint(m,n,k)`	m, m+k, m+2k, ..., m+q*k (where q is the largest integer such that $m + q * k \leq n - k$)	`rint(3,12,3)` = 3, 6, or 9	Returns a random integer between m and n-1 (inclusive) in steps of k. Generates approximately (n-m)/k variations.
`range(n)`	1, ..., n	`range(3)` = 1, 2, or 3	Returns a random integer between 1 and n (inclusive). Generates n variations.
`range(m,n)`	m, ..., n	**range(3,6)** = 3, 4, 5, or 6	Returns a random integer between m and n (inclusive). Generates (n-m)+1 variations.
`range(m,n,k)`	m, m+k, m+2k, ..., m+q*k (where q is the largest integer such that $m + q * k \leq n$)	`range(3,12,3)` = 3, 6, 9, or 12	Returns a random integer between m and n (inclusive), in steps of k. Generates approximately (n-m)/k +1 variations.
`rand(m,n)`	m, ..., n	`rand(0.5,9.5)` = all real numbers between 0.5 and 9.5 (inclusive)	Returns a random real number between m and n (inclusive).

Function	Range of Operation	Example	Description
`rand(m,n,k)`	m, ..., n expressed to k significant figures	**rand(0.5,9.5,3)** = all real numbers between 0.5 and 9.5 (inclusive), expressed to 3 significant figures	Returns a random real number between m and n (inclusive), *expressed to* k *significant figures*

For details on each of these functions, see *Functions and Arguments (page 222)*.

In addition to Maple T.A.'s built-in functions, you can access Maple™ commands. Maple provides many randomization commands. The following table lists selected randomization commands in Maple. For more information on Maple functions that can be used to generate random numbers, refer to your Maple documentation.

Table 10.10: Maple Randomization Commands

Maple Function	Use
rand	generate a random 12-digit non-negative integer, or a random integer within a specified range (inclusive)
randpoly	random polynomial generator
LinearAlgebra[RandomMatrix]	construct a random Matrix
LinearAlgebra[RandomVector]	construct a random Vector
RandomTools[Generate]	generate a particular random object (For example, return a random complex number with real and imaginary parts of a specified flavor, or choose one of the entries in a collection with equal probability)

Note: When using Maple's random number generator **rand**, you must include **randomize():** as the first part of the call. This sets the initial state of the random number generator using a number based on the system clock instead of the default seed in Maple. Without **randomize()**, each call to **rand** will produce the same sequence of values using the default randomization seed. For an example that uses Maple randomization commands, see *Tutorial: Maple-based Algorithmic Question Authoring (page 244)*.

See Also:

Tutorial: Algorithmic Question (page 241)

Tutorial: List-based Variables (page 242)

Tutorial: Maple-based Algorithmic Question Authoring (page 244)

10.18 Functions and Arguments

The system contains numerous built-in functions for creating algorithmic variables. You can create random integers or real numbers, perform operations on lists of items, or set conditions on variables.

This section describes the built-in functions, and gives the syntax and purpose of each function.

Most of these functions are for use in authoring questions, but cannot be used in student responses. For more information, see *Mathematical and Logical Operators in Algorithms and Answers (page 233)*.

condition:x

`condition:x`

Imposes the condition defined by statement **x**, which is typically constructed using one or more other functions.

For example,

```
$a=range(-10,10);
condition: ne($a,0);
```

This generates a random non-zero integer between -10 and 10 (inclusive).

decimal(n, x), sig(n, x), int(x)

decimal(n, x)

Returns **x** expressed as a floating-point number rounded to **n** decimal places. When decimal(0, expression) is used with an expression, the system automatically inserts commas to separate thousands.

Trailing zeros are truncated and not displayed. For another option, see *numfmt(fmt, x) (page 227)*.

sig(n, x)

Returns **x** expressed as a floating-point number rounded to **n** significant digits. In cases of possible ambiguity, scientific notation is used to display the value.

int(x)

Returns the integer part of **x.**

For example,

decimal(3, 20.8571)

returns **20.857**

sig(3, 20.8571)

returns **20.9**

```
int(20.8571)

returns
 20
```

eq(a, b), ge(a, b), le(a, b), ne(a, b)

eq(a, b)
Returns 1.0 if **a** and **b** are equal. Otherwise, it returns 0.0.

ge(a,b)
Returns 1.0 if **a** is greater than, or equal to **b**. Otherwise, it returns 0.

le(a,b)
Returns 1.0 if **a** is less than, or equal to **b**. Otherwise, it returns 0.0.

ne(a,b)
Returns 1.0 if **a** and **b** are not equal. Otherwise, it returns 0.0.

For example,

```
if(eq($a, $b), "Red", "Green")
```

returns **Red** if **$a=$b** and **Green** otherwise.

```
if(ge($a, $b), "Red", "Green")
```

returns **Red** if **$a>=$b** and **Green** otherwise.

```
if(le($a, $b), "Red", "Green")
```

returns **Red** if **$a<=$b** and **Green** otherwise.

```
if(ne($a, $b), "Red", "Green")
```

returns **Red** if **$a** and **$b** are not equal and **Green** otherwise.

See Also:

if(a, b, c) (page 225)

fact(n)

```
fact(n)
```

Returns factorial **n**.

If **n** is not an integer returns **fact(int(n))**.

If **n** is negative, returns 1.

For example,

```
fact(4)
```

returns 24

frac(a, b)

frac(a, b)

The **frac(a,b)** structure typesets x/y as a fraction in its lowest terms. For example, the fractions "**4/8**" and "**2/6**" are presented as "**1/2**" and "**1/3**", in simplest form.

frac(a,b)returns a string that expresses the fraction **a/b** in its lowest terms.

For example,

```
frac(12, 15)
```

returns **4/5**

```
frac(12, 3)
```

returns **4**

Note: frac(a,b) can be combined with **mathml(s)** to produce nicely typeset fractions. For example,

```
mathml(frac(8, 12))
```

displays

$$\frac{2}{3}$$

gcd(a, b)

gcd(a, b)

Returns the greatest common divisor of a and b.

For example,

```
gcd(12, 15)
```

returns **3**

gt(a, b), lt(a, b)

```
gt(a, b)
```

Returns 1.0 if **a** is greater than **b,** that is **(a > b)**. Otherwise, it returns 0.0.

```
lt(a, b)
```

Returns 1.0 if **a** is less than **b,** that is **(a < b)**. Otherwise, it returns 0.0.

if(a, b, c)

```
if(a, b, c)
```

If **a** is nonzero, it returns **b**. Otherwise, it returns **c**.

indexof(k, a, b, c, d, ...)

Returns the index of an item within a list (the first item is in position 0, the second in position 1, and so on).

For example,

indexof(3, 2, 3, 5, 7, 11)

returns **1**

java(cn, a, b, c, d, ...)

java(cn, a, b, c, d, ...)

Pass the arguments a,b,c,d,.. to a custom Java™ evaluation engine and return the result. The first argument (cn) must be a string giving the fully qualified name of a Java class that implements the interface

```
gateway.question.random.AlgorithmicFunction.
```

This interface has a single public method.

```
public String eval(String[ ] args);
```

The arguments a,b,c,d,... are passed to eval() in a string array.

For example,

```
java("com.mycompany.QuoteFunction", "SUNW")
```

returns a real-time quote for Sun Microsystems, Inc. stock (assuming that the class QuoteFunction had been suitably programmed).

lsu(n, x)

```
lsu(n, x)
```

Returns the unit in the **n**th significant place of **x**. (This is called the least significant unit of **x** in the **n**th place).

For example,

lsu(3, 3.14159)

returns **0.01** (the unit in the third significant place).

This operation is designed to be used when setting the tolerance for correct answers. For example, to accept an answer to within a tolerance of one unit in the third significant digit, use:

```
$ans  = <formula>;
$tol = lsu(3, $ans);
```

and set the answer field to:

```
$ans ? $tol
```

maple(text)

Pass text to the Maple™ kernel and return the value of the last line processed.

For example,

maple("ithprime(12)")

returns **37**

```
maple("diff(sin(x)*x, x)")
```

returns **cos(x)*x + sin(x)**

See Also:

Mathematical and Logical Operators in Algorithms and Answers (page 233)

mathml(f)

Returns a string consisting of the formula **f** typeset in MathML.

Note: The system renders MathML; the MathML tags do not appear on the screen when the value of the string is displayed. Instead, the rendering appears.

For example,

```
mathml("x^$n")
```

returns

x^n

By default, **mathml(f)** automatically simplifies the formula. To prevent auto-simplification, use the "**nosimplify**" option.

For example,

```
mathml("x^2 + 4/8", "nosimplify")
```

returns

$$x^2 + \frac{4}{8}$$

If you want to include the function in another function, use the "**notags**" option.

```
$\sum0^\infty\var{mathml("1/$a x^2  + $b)", "notags")}$
```

See Also:

Inline Display of Formatted Math Expressions (page 196)

Simplified Math Expressions (page 199)

max(a, b, c, d, ...), min(a, b, c, d, ...)

max(a, b, c, d, ...)

Returns the largest of the arguments.

For example,

```
max(3,5,7,4.8,-1)

returns
  7
```

min(a, b, c, d, ...)

Returns the smallest of the arguments.

For example,

```
min(3, 5, 7, 4.8, -1)

returns
  -1
```

not(a)

not(a)

Returns 1.0 if **a** is equal to 0.0. Otherwise, it returns 1.0.

numfmt(fmt, x)

Returns the value of **x**, formatted according to the template **fmt**.

For example,

numfmt("#.00", 20.9)

returns **20.90**.

plotmaple(text)

Maple T.A. uses Maple™ plotting features to display plots. Maple provides many ways of representing data and mathematical expressions graphically using plots. **You can display a plot with any question type.**

The **plotmaple** command can be used anywhere an algorithmic variable can be used, that is, in text, hints, feedback, and more.

```
plotmaple("plotstatement, libname='filename.lib', plotdevice='gif|jpeg',
plotoptions='options'");
```

Where

- **plotstatement -** plot expression and plot options
- **libname** (optional) - specifies path to user Maple libraries
- **plotdevice** (optional) - gif (produces a GIF/ANIMATED-GIF) or jpeg (produces a 24-bit color JPEG rendering of the image)
- **plotoptions** (optional) - specifies plot options available for the plot device specified (see **Table 10.11**)

Some examples:

```
plotmaple("plot3d(sin(x)*y, x=0..10, y=-1..1, style=patchnogrid, lightmodel=light1),
plotdevice='jpeg', plotoptions='width=250,height=250'")

plotmaple("plot(sin(x), x=-Pi..Pi), plotdevice='gif', plotoptions='height=250, width=250'");


plotmaple("plot(sin(x), x=-Pi..Pi), plotdevice='jpeg', plotoptions='height=250, width=250'");
```

Table 10.11: Options for GIF and JPEG Plots

GIF plot options	**result**
colors=n	Set the number of colors. The default 256 colors.
height=n	Set the number of pixels in the height. The default is 512.
width=n	Set the number of pixels in the width. The default is 512.
JPEG plot options	**result**
height=n	Specify the height of the canvas. The default is 360.
quality=n	Set the JPEG quality level, an integer value from 0 to 100. The default is 95. Settings greater than 95 generally have negligible visible effect.
width=n	Specify the width of the canvas. The default is 480.

See Also:

Setting Plot Options (page 235)

rand(m, n), rand(m, n, k)

```
rand(m,n)
```

Returns a random real number between **m** and **n** (inclusive). For example, to generate random numbers between **0.5** and **9.5**, use the formula:

rand(0.5, 9.5)

```
rand(m,n,k)
```

Returns a random real number between **m** and **n** expressed to **k** (**k>0**) significant digits (**k** is truncated to an integer). For example, to generate numbers between **2.73** and **7.91** to two significant figures, use:

rand(2.73, 7.91, 2)

Note: In this definition, the dummy variable **k** represents the number of significant figures, not the step size (as it indicates in defining **range** and **rint**).

To generate random numbers that vary an order of magnitude and have the same number of decimal digits, use the **decimal** function in conjunction with the **rand** function. For example, to generate a random number between **2** and **20** with one decimal place, use:

decimal(1, rand(2, 20, 3))

Note: Numbers posing a possible ambiguity are returned using scientific notation.

See Also:

decimal(n, x), sig(n, x), int(x) (page 223)

Random Integer Generation range(n), range(m, n), range(m, n, k) (page 229)

Random Integer Generation rint(n), rint(m, n), rint(m, n, k) (page 230)

Generating Random Numbers in Questions (page 221)

Random Integer Generation range(n), range(m, n), range(m, n, k)

The **range** function return random integers.

`range(n)`

The **range(n)** function generates a random integer in the range **1, ..., n** (inclusive). There are **n** numbers in that range. Hence, it is a selection of one number from a choice of **n**.

`range(m,n)`

The **range(m,n)** function generates a random integer in the range **m, ..., n** (inclusive).

`range(m,n,k)`

The **range(m,n,k)** function generates a random integer in the range **m, m+k, ..., m+q*k** (inclusive) where **q** is the largest integer such that **m+q*k<=n**.

That is,

```
range(n) = range(1,n) = range(1,n,1)
```

Examples

range(3) returns **1**, **2**, or **3**.

range(2,3) returns **2** or **3**.

range(0,7,3) returns **0**, **3**, or **6**.

range(0,8,3) returns **0**, **3**, or **6**.

range(0,9,3) returns **0**, **3**, **6**, or **9**.

Note: The **rint** function returns random numbers in the range **0, ..., n-1** (inclusive).

See Also:

rand(m, n), rand(m, n, k) (page 228)

Random Integer Generation rint(n), rint(m, n), rint(m, n, k) (page 230)

Generating Random Numbers in Questions (page 221)

rank(n, a, b, c, d, ...)

Returns the **n**th largest element item from a list. (Numbering starts at 1)

For example,

rank(3, 2, 4, 6, 5, 7)

returns **5**

See Also:

indexof(k, a, b, c, d, ...) (page 225)

switch(n, a, b, c, ...) (page 232)

Random Integer Generation rint(n), rint(m, n), rint(m, n, k)

The **rint** function return random integers.

```
rint(n)
```

The **rint(n)** function generates a random integer in the range **0, ..., n-1** (inclusive). There are **n** numbers in that range. Hence, it is a selection of one number from a choice of **n**.

```
rint(m,n)
```

The **rint(m,n)** function generates a random integer in the range **m, ..., n-1** (inclusive).

```
rint(m,n,k)
```

The **rint(m,n,k)** function generates a random integer in the range **m, m+k, ..., m+q*k** (inclusive) where **q** is the largest integer such that **m+q*k<=n-k**.

That is,

```
rint(n) = rint(0,n) = rint(0,n,1)
```

Examples

rint(3) returns **0**, **1**, or **2**.

rint(1,3) returns **1** or **2**.

rint(0,7,3) returns **0** or **3**.

rint(0,8,3) returns **0** or **3**.

rint(0,9,3) returns **0**, **3**, or **6**.

Note: The **range** function returns random integers in the range **0, ..., n** (inclusive).

See Also:

Random Integer Generation range(n), range(m, n), range(m, n, k) (page 229)

rand(m, n), rand(m, n, k) (page 228)

Generating Random Numbers in Questions (page 221)

Statistical Operations

binomial(n, r)

Computes the **r**th binomial coefficient of degree **n**. This corresponds to the number of ways of choosing **r** objects from a set of **n**, ignoring order. It is used in system performance statistics, but is also available for use in internal directives (algorithms and answers).

For example,

binomial(22, 3)

returns the coefficient of x^{19} in $(x+1)^{22}$ or 1,540.

erf(z)

Computes the cumulative probability (probability that a variate assumes a value less than or equal to **z**) for a standard normal distribution (that is, with mean **0** and variance **1**).

For example,

```
erf(0) = 0.5
erf(1) = 0.68
erf(2) = 0.95
```

erf(3) = 0.997

Other normal distributions can be modeled by re-normalizing (that is, **erf((z-m)/s)** is the probability distribution of a normal distribution with mean **m** and standard deviation **s**).

It is used in producing performance statistics within the system, but is also available for use in internal directives (algorithms and answers).

inverf(p)

The inverse function of **erf(z)**: Let Z be a standard normally distributed random variable. Then **inverf(p)** is the value of **z** for which **P(Z<z) = p**

For example,

inverf(0.99)

returns the value of **z** with probability 0.99 that **Z<z**.

invstudentst(k, x)

Computes the inverse of the cumulative probability density of the Students-t distribution with k degrees of freedom.

For example,

invstudentst(2, 0.9)

returns the value of **x** with probability equal to **0.9**.

studentst(k, x)

Compute the cumulative probability distribution at x of the Students-t distribution with **k** degrees of freedom.

For example,

studentst(2, 1.55)

returns the probability that **X<1.55.**

strcat(a, b, c, d, ...)

strcat(a, b, c, d, ...)

Returns the concatenation of the strings in the list.

For example

```
strcat("$a", " and ", "$b")
```

returns **"cats and dogs"**

when $a="cats" and $b="dogs"

sum(varname, start, stop, expr)

sum(varname, start, stop, expr)

Sums the expression (**expr**) with respect to the dummy variable (**varname**) between the values **start** and **stop**.

For example,

```
sum(i, 1, 20, i^2)
```

evaluates **1^2** + **2^2** + **3^2** + **...** + **20^2** and returns **2,870**.

```
$s = sum(i, 1, $n, i^2);
```

calculates the sum of the squares of the integers from **1** to **n**.

Note: Do not use a **$** character for the dummy variable in the summation.

switch(n, a, b, c, ...)

Given a whole number **0**, **1**, **2**, ... and a list of items, returns the **n**th item in the list (the first item is in position **0**, the second in position **1**, and so on).

For example,

```
$prime=switch(rint(5), 2, 3, 5, 7, 11)
```

sets **$prime** to a random choice from the first 5 primes.

switch(1, "red", "green", "yellow")

returns **green**

See Also:

Tutorial: List-based Variables (page 242)

rank(n, a, b, c, d, ...) (page 230)

Random Integer Generation rint(n), rint(m, n), rint(m, n, k) (page 230)

10.19 Mathematical and Logical Operators in Algorithms and Answers

This is a list of mathematical and logical operators (functions) that you can use in internal directives within questions. Some of these operators are also available to students in their responses to math questions in the system (unreserved operations), while others are reserved for instructor or question-authoring use only.

Unreserved Operations

These functions are available for use in both internal directives (in variable definitions and statements) and for use by students when expressing their responses to math questions. Each command performs the expected mathematical function in a formula or algorithm.

Table 10.12: Mathematical Functions that Students Can Use

sin, cos, tan	**arcsin, arccos, arctan**
csc, sec, cot	**hypsin, hypcos, hyptan**
abs, sqrt	**archypsin, archypcos, archyptan**
log, ln, exp	

Reserved Operations

The following functions are only available as internal system directives (in variable definitions and statements). These functions are not available to students.

Table 10.13: Mathematical Functions Reserved for Instructor Use

binomial, erf, inverf, invstudentst, studentst	**indexof**	**plotmaple**
decimal, sig, int	**java**	**rand**
eq, ge, le, ne	**lsu**	range
fact	**maple**	**rank**
frac	**mathml**	**rint**
gcd	**max, min**	**strcat**
gt, lt	**not**	**sum**
if	**numfmt**	**switch**

See Also:

Mathematical Functions & Operations (page 204)

10.20 Question Chaining Variables

Question chaining is an authoring technique in which you create reference objects in one question that are called into display in the next question. By using question chaining, you can capture a student's response to the first question in an assignment, and then include that response as part of other questions in the same assignment.

To use question chaining, you must:

1. *Author* questions that use the special variables **$response** and/or **$grade** described below.
2. *Create assignments* that use these questions in sequence.

Note: The questions that use chaining must have an algorithm. If this is not needed, you can define a dummy variable. This is demonstrated in the example. See the line **qu.1.2.algorithm=$x=0;@**

To use question chaining variables:

- If a question has an **id** field set, then you can use the **id** tag to refer to the response the student provided during an assignment.

Example

```
qu.1.topic=Question Chaining@

qu.1.1.mode=Blanks@
qu.1.1.id=foobar@
qu.1.1.editing=useHTML@
qu.1.1.question=Pick a number between 1 and 7: <1> @
qu.1.1.blank.1=1, 2, 3, 4, 5, 6, 7@
qu.1.1.grader.1=formula@
qu.1.1.extra=@

qu.1.2.mode=Numeric@
qu.1.2.editing=useHTML@
qu.1.2.question=What is the sum of $response.foo1 and 17?@
qu.1.2.algorithm=$x=0;@
qu.1.2.answer.num=${($response.foobar + 17)}@
qu.1.2.answer.units=@
qu.1.2.showUnits=false@
qu.1.2.grading=exact_value@
qu.1.2.name=Exact answer Precision with system units@
qu.1.2.editing=useHTML@qu.1.1.mode=formula@
```

To refer to the student response to one question in another question, use the variable **$response.foobar**. The referring questions need not be in the same question bank, but they must use the correct ID code. The contents of **$response.foobar** is a string.

Note: $response.foobar can be used only in in-line dynamic text; it cannot be used in the algorithm section of a question because the algorithm is initialized before the assignment is started.

Questions frequently have multipart responses. For example, suppose **foobar** is a **Blanks** question with 5 blanks. You can access the blanks individually using **$response.foobar.1**, .., **$response.foobar.5**. In the case of a multipart question with parts (a),..(c), each of which holds a **Blanks** question with 5 blanks, you can access the responses using **$response.foobar.1.1**, ..., **$response.foobar.3.5**. You can have as many levels of depth as are needed.

In addition to accessing the raw text of the student response, you can also access the grade that is assigned to the current response, using **$grade.foobar** (or, as appropriate, **$grade.foobar.3.5**). This is a decimal number between 0.0 and 1.0. In the case that the question has not yet been answered, the grade is -1.0. (You should be careful revealing grade information, because that may undermine the intent of the instructor in setting the assignment.)

Note: $response and **$grade** can be used only in in-line expressions. You can use these variables inside only question statements and answer definitions and **not** inside algorithm statements because the **algorithm** field of the question is evaluated only once, when the question is first loaded at the time of initial assignment creation. To use the **response** object inside other variable calculations would require that the **algorithm** field be refreshed each time the question is presented, which would incorrectly update the other variable values in the **algorithm** field.

Example:

```
qu.1.3.mode=Blanks@
qu.1.3.id=foo2@
qu.1.3.editing=useHTML@
qu.1.3.question=Pick a number between 1 and 7:  <1> @
qu.1.3.blank.1=1, 2, 3, 4, 5, 6, 7@
qu.1.3.grader.1=formula@
qu.1.3.extra=@
qu.1.4.mode=Numeric@
qu.1.4.editing=useHTML@
qu.1.4.question=What is the sum of $response.foo2 and 17?@
qu.1.4.algorithm=
@
qu.1.4.answer.num=${($response.foo2 + 17)}@
qu.1.4.answer.units=@
qu.1.4.showUnits=false@
qu.1.4.grading=exact_value@
qu.1.4.name=Exact answer Precision with system units@
qu.1.4.editing=useHTML@
```

Question: How do I use the **$grade** variable to return a statement that tells the student on the following question "Your response was correct/incorrect" depending on the grade in the first question?

Answer: Try

```
${if(eq($grade.foo2, 1), "correct", "incorrect")}
```

10.21 Setting Plot Options

2-D Plot Statement Options

The options to the Maple **plot** command are given after the function(s), horizontal range, and vertical range, as equations of the form option = value.

For example,

```
plotmaple("plot(sin(x), x=-Pi..Pi, filled=true, title=`My Plot`)");
```

The options described below are supported.

Table 10.14: Options for 2-D Plots

Option	Description
adaptive	When plotting a function over an interval, the interval is sampled at a number of points, controlled by **sample** and **numpoints**. Adaptive plotting, where necessary, subdivides these intervals to attempt to get a better representation of the function. This subsampling can be turned off by setting the **adaptive** option to **false**. By default, this option is set to **true**, and intervals are subdivided at most 6 times in trying to improve the plot. By setting this option to a non-negative integer, you can control the maximum number of times that subintervals are divided.
axes=f	Specifies the type of axes, one of: **boxed**, **frame**, **none**, or **normal**. The value can be in lowercase or uppercase.
color=n or **colour=n**	Allows the user to specify the color of the curves to be plotted.
coords=<name>	Indicates that a parametric plot is in the coordinate system specified by **<coord_name>**.
discont=s	Setting **s** to **true** forces **plot** to call the function **discont** first to determine the discontinuities of the input (or **fdiscont** if **discont** fails or if the input is not an expression) and then breaks the horizontal axis into appropriate intervals where the expression is either continuous or contains only removable discontinuities.

Option	Description
filled=truefalse	If the **filled** option is set to **true**, the area between the curve and the x-axis is given a solid color. This option is valid only with the following commands: **plot**, **contourplot**, **implicitplot**, **listcontplot**, **polarplot**, and **semilogplot**.
labels=[x,y]	This option specifies labels for the axes. The values of **x** and **y** must be strings. The default labels are the names of the variables in the original function to be plotted, if any. The strings must be entered using left single quotes.
labeldirections=[x,y]	This option specifies the direction in which labels are printed along the axes. The values of **x** and **y** must be **horizontal** or **vertical**. The default direction of any labels is **horizontal**. The value can be in lowercase or uppercase.
legend=s	A legend for a plot can be specified by either a string or a list of strings. When more than one curve is being plotted then they must be specified as a list (and not a set) and there must be a legend for each curve where the *i*th curve is associated with the *i*th legend. The strings must be entered using left single quotes.
linestyle=n	Controls the dash pattern used to render lines in the plot. The **linestyle** can be specified as either an integer between 1 and 4, or a name from the following list: **SOLID**, **DOT**, **DASH**, or **DASHDOT**. The value must be in uppercase.
numpoints=n	Specifies the minimum number of points to be generated. The default is **50**. **Note: plot** employs an adaptive plotting scheme which automatically does more work where the function values do not lie close to a straight line. Hence, **plot** often generates more than the minimum number of points.
resolution=n	Sets the horizontal display resolution of the device in pixels (the default is **n = 200**). The value of **n** is used to determine when the adaptive plotting scheme terminates. A higher value results in more function evaluations for non-smooth functions.
sample	A list of values which is to be used for the initial sampling of the function(s). When coupled with **adaptive=false**, this option allows explicit control over the function evaluations performed by **plot**.
scaling=s	Controls the scaling of the graph as **constrained** or **unconstrained**. The default value is **unconstrained**. The value can be in lowercase or uppercase.
style=s	The interpolation style must be one of **line**, **patch**, **patchnogrid**, or **point**. The default is **line**. The value can be in lowercase or uppercase. The **point** style plots points only, **line** interpolates between the points, **patch** uses the patch style for plots containing polygons, and **patchnogrid** is the **patch** style without the grid lines.
symbol=s	Symbol for points in the plot, where the value **s** is one of **box**, **circle**, **cross**, **diamond**, or **point**. The value can be lowercase or uppercase. **Note:** If the style is set to **point**, the default symbol is plot device specific.
symbolsize=n	The size (in points) of a symbol used in plotting can be given by a positive integer. This does not affect the symbol **POINT**. The default symbol size is **10**.
thickness=n	This option specifies the thickness of lines in the plot. The **thickness n** must be a non-negative integer. The default thickness is **0**.
tickmarks=[m,n]	If **m** is an integer, then this option controls the number of tickmarks that are placed on the x-axis. If **m** is a list of numeric values, then tickmarks are placed only at the specified values. If **m** is a list of equations **v=t**, where **v** is a numeric value and **t** is a name or string, then a tickmark with the label **t** is placed at **v**. If **m** is the name **default**, then the default tickmark placement is applied. Similarly, **n** specifies the placement of y-axis tickmarks. This option can be replaced by the **xtickmarks=m** or **ytickmarks=n** option if you wish to specify tickmarks for one axis only.
title=t	The title for the plot. The value **t** must be a character string. The default is no title. You can create multi-line titles for standard plots. Use the characters **\n** in the character string to denote the start of a new title line. The strings must be entered using left single quotes.

Option	Description
view=[xmin..xmax, ymin..ymax]	This option indicates the minimum and maximum coordinates of the curve to be displayed on the screen. The default is the entire curve.
xtickmarks=m	See the description for the **tickmarks** option.
ytickmarks=n	See the description for the **tickmarks** option.

See Also:

plotmaple(text) (page 227)

3-D Plot Statement Options

Options to the Maple **plot3d** function are given after the first three arguments, as equations of the form option = value.

For example,

```
plotmaple("plot3d(sin(x)*y, x=-Pi..Pi, y=-1..1, lightmodel=light1)");
```

The options described below are supported.

Table 10.15: Option for 3-D Plots

Option	Description
ambientlight=[r, g, b]	This option sets the red, green, and blue intensity of the ambient light for user defined lighting. The values of **r**, **g**, and **b** must be numeric in the range 0 to 1.
axes=f	This option specifies how the axes are to be drawn, where **f** is one of **boxed**, **normal**, **frame**, or **none**. The default value is **none**. The value can be in lowercase or uppercase.
color=c or **colour=c**	This option defines a color value or function. Note that **color** specification takes precedence over **shading** specification, regardless of the order of the options in the command.
contours=n	This option specifies the number of contours or a list of contour values, where **n** is a positive integer or a list of contour values. The default value is **10**.
coords=c	This option specifies the coordinate system to be used. The default is **Cartesian**.
filled=truefalse	If the **filled** option is set to **true**, the region between the surface and the xy-plane is displayed as solid. This option is valid only with the following commands: **plot3d**, **contourplot3d**, and **listcontplot3d**.
grid=[m,n]	This option specifies the dimensions of a rectangular grid on which the points are generated (equally spaced).
gridstyle=x	This option specifies a grid **x** as either **rectangular** or **triangular**. This option is not available for all 3-D plot objects.
labeldirections=[x,y,z]	This option specifies the direction in which labels are printed along the axes. The values of **x**, **y**, and **z** must be **horizontal** or **vertical**. The default direction of any labels is **horizontal**. The value can be in lowercase or uppercase. The **axes** option must be set to **boxed**, **frame**, or **normal**.
labels=[x,y,z]	This option specifies labels for the axes. The value of **x**, **y**, and **z** must be a string. The default labels are the names of the variables in the original function to be plotted. The strings must be entered using left single quotes.
light=[phi, theta, r, g, b]	This option adds a directed light source from the direction **phi**, **theta** in spherical coordinates with red, green, and blue intensities given by r, g, and b. The values of **r**, **g**, and **b** must be numeric in the range 0 to 1.
lightmodel=x	This option chooses a predefined light model to illuminate the plot. Valid light models include **none**, **light1**, **light2**, **light3**, and **light4**. The value must be in lowercase.

Option	Description
linestyle=n	This option controls the dash pattern used to render lines in the plot. The linestyle can be specified as either an integer between 1 and 4, or a name from the following list: **SOLID**, **DOT**, **DASH**, and **DASHDOT**. The value must be in uppercase.
numpoints=n	This option specifies the minimum total number of points to be generated (default 625 = 25^2). The **plot3d** command uses a rectangular grid of dimension **isqrt(n+3)+1**.
orientation=[theta, phi]	This option specifies the **theta** and **phi** angles of the point in three dimensions from which the plot is to be viewed. The default is at a point that is out perpendicular from the screen (negative **z** axis) so that the entire surface can be seen. The point is described in spherical coordinates where **theta** and **phi** are angles in degrees, with default 45 degrees in each case.
projection=r	This option specifies the perspective from which the surface is viewed, where **r** is a real number between 0 and 1. The 1 denotes orthogonal projection, and the 0 denotes wide-angle perspective rendering. **r** can also be the one of the names, **fisheye**, **normal**, or **orthogonal**, which correspond to the projection values 0, 0.5, and 1, respectively. The default is **orthogonal**. The value can be in lowercase or uppercase.
scaling=s	This option specifies whether the surface should be scaled so that it fits the screen with axes using a relative or absolute scaling, where **s** is either **unconstrained** or **constrained**. The default is relative **unconstrained**. The value can be in lowercase or uppercase.
shading=s	This option specifies how the surface is colored, where **s** is one of **xyz, xy, z, zgrayscale**, **zhue**, or **none**. The default is device dependent. The value can be lowercase or uppercase. Note that **color** specification takes precedence over **shading** specification, regardless of the order of the options in the command. The value can be in lowercase or uppercase.
style=s	This specifies how the surface is to be drawn, where **s** is one of **contour**, **hidden**, **line**, **patch**, **patchcontour**, **patchnogrid**, **point**, or **wireframe**. Some of the options may not be available on some devices. The default style is **patch** (for colored surface patch rendering). The value can be in lowercase or uppercase.
symbol=s	This option defines the symbol for points in the plot. **s** is one of **box**, **circle**, **cross**, **diamond**, or **point**. The value can be in lowercase or uppercase. **Note:** If the style is set to **point**, the default symbol is plot device specific.
symbolsize=n	The size (in points) of a symbol used in plotting can be given by a positive integer. This does not affect the symbol **point**. The default symbol size is **10**.
thickness=n	This option specifies the thickness of lines in the plot. The **thickness n** must be a non-negative integer. The default thickness is **0**.
tickmarks=[m,n,p]	If **m** is an integer, then this option controls the number of tickmarks that are placed on the x-axis. If **m** is a list of numeric values, then tickmarks are placed only at the specified values. If **m** is a list of equations **v=t**, where **v** is a numeric value and **t** is a name or string, then a tickmark with the label **t** is placed at **v**. If **m** is the name **default**, then the default tickmark placement is applied. Similarly, **n** specifies the placement of y-axis tickmarks and **p** specifies the placement of z-axis tickmarks.
title=t	This option specifies a title for the plot. The value of **t** must be a string. The default is no title. You can create multi-line titles for standard plots. Use the characters **\n** to denote the start of a new title line. The strings must be entered using left single quotes.
view=zmin..zmax or **[xmin..xmax, ymin..ymax, zmin..zmax]**	This option indicates the minimum and maximum coordinates of the surface to be displayed on the screen. The default is the entire surface.

See Also:

plotmaple(text) (page 227)

10.22 Details on Using Algorithmic Variables

The Answer Field Evaluation and Requirements

The **answer** field specifies internal directives, or instructions to the system for how to grade the question. It operates like a command-line statement when processed by the system.

The **answer** field accepts only values that can be system-evaluated: numbers, formulas, algorithmic variables, or numbers and/or formulas with required physical units.

The system recognizes:

- Equivalent numeric and algebraic expressions
- Equivalent standard physical units
- Scientific notation
- The **?** operator as a representation of a margin of error. The answer:

(3.75 ? 0.05) kg
grades all responses in the range **3.70** to **3.80 kg** (inclusive) correct.

> **Note:** You cannot use the **?** operator in the numeric answer (**answer.num** field) for a numeric question. You must use the specialized tolerance fields.

Variable definitions must reduce to one of the above.

You can round a variable value to any number of significant digits, and then specify the variable as the answer.

Controlling Display of Numbers within Questions

All random numbers generated by the system from variable definitions are floating-point variables**.** By default, they are expressed to about **8** significant digits.

To control the display of numeric variables, enclose expressions in an **int()** statement to make them integers (for example, **$m=int(14.5*5);**), or in double quotes **""** to make them strings (for example, **$s="380";**).

You can also control the appearance of numeric variables by enclosing random number generating statements and arithmetic operations with **decimal(n,(statement))**, to express the result to **n** decimal places, or by using the **sig(n,(statement))** structure to express the result to **n** significant digits.

See Also:

Random Integer Generation rint(n), rint(m, n), rint(m, n, k) (page 230)

decimal(n, x), sig(n, x), int(x) (page 223)

Variables within Variable Statements

You can use variables as arguments in variable definition statements.

Since the algorithm generation works linearly, you must define variables before referencing them in other variable definition statements (or control sequences).

Example

```
qu.3.7.mode=Formula@
qu.3.7.algorithm=
      $n=rint(8);
      $x=rand(1,10);
      $ans=sig($n,  $x)
@
```

```
qu.3.7.question=
  Express the number $x to $n significant places.
@
qu.3.7.answer=$ans@
```

See Also:

Controlling Answer Tolerance (page 183)

decimal(n, x), sig(n, x), int(x) (page 223)

Formula (page 137)

rand(m, n), rand(m, n, k) (page 228)

Random Integer Generation rint(n), rint(m, n), rint(m, n, k) (page 230)

Displaying the $ (Dollar Sign) in Questions

The use of the **$** is reserved for variable definitions in algorithms in the Maple T.A. question syntax.

To display a **$** character in the text of your question, use:

```
\$
```

Displaying the Backslash Symbol in Questions

The use of the backslash character is reserved in the Maple T.A. system question syntax. To include a \ character in the text of your question, use:

```
\\
```

Text Strings in Variable Statements

Statements defining variables can also include text strings. Strings must be enclosed in double quotes **""**.

Example

```
qu.4.1.mode=Formula@
qu.4.1.algorithm=
       $animal=switch(rint(2),  "cat", "dog");
       $mass=if(int(eq("$animal",  "cat")), rand(2,15,2), rand(10,30,2));
       $velocity=rand(2,5,2)
@
qu.4.1.question=
A $mass kg $animal is running at $velocity m/s. What is its kinetic energy?
@
qu.4.1.answer= (1/2)$mass * $velocity^2
@
```

In the test for the type of animal in the definition of **$mass**, it is required that the variable **$animal** be enclosed in quotes.

Note: The strings in the example above appear only as arguments to functions. It is recommended that you do **not** mix strings and numbers in expressions.

10.23 Tutorial: Algorithmic Question

Algorithmically generated variables are used in questions to generate random numbers in questions and create multiple permutations of questions from a single template. The system uses a special field (**algorithm**) to contain variable definitions. Variable definitions must be presented in the proper system syntax. They can contain embedded functions and arguments.

Note: For any question type, you can use Maple commands and Maple T.A. to generate algorithmic numbers. For more information, see *Tutorial: Maple-based Algorithmic Question Authoring (page 244)*

This tutorial examines the process of converting a typical numeric question to an algorithmically generated question by using variables.

Consider a single example. Substitute variable data for the mass and speed values in the following question each time the assignment is generated.

A ball of mass 0.8 kg is thrown at a speed of 12.6 m/s. (a) What is its momentum? (b) What is its kinetic energy?

The plain-text script for this question with no algorithms is:

```
qu.1.1.mode=Multipart Formula@
qu.1.1.question=
A ball of mass 0.8 kg is thrown at a speed of 12 m/s.
<br>(a) What is its momentum?
<br>(b) What is its kinetic energy?
@
qu.1.1.answer=
9.6 m/s; 46.08 J
@
```

The plain-text script for this question, created algorithmically, is:

```
qu.1.1.mode=Multipart Formula@
qu.1.1.algorithm=
  $m=decimal(1,rand(0.5,  1.0));
  $v=decimal(0,rand(10,  15));
  $ansa=decimal(2,($m*$v));
  $ansb=decimal(2,(1/2)*$m*$v^2);
  @
qu.1.1.question=
A ball of mass $m kg is thrown at a speed of $v m/s.
<br>(a) What is its momentum?
<br>(b) What is its kinetic energy? @
qu.1.1.answer= $ansa; $ansb@
qu.1.1.comment=
  $ansa kg m/s is its momentum.<br>
  $ansb J is its kinetic energy.
  @
```

Note that the **algorithm** field and variable definitions have been added.

- The variables are represented by alphanumeric names that must start with a **$** followed by an alphanumeric sequence that consists of at least one letter.
- The first character in the name of the variable must be a letter (not a number).
- Variables are defined in a series of statements resembling equations of the form:

```
$variable=<formula>;
```

- Each variable definition is separated by a semicolon (the final semicolon in a series of variable definitions is optional).
- The syntax of the formula is similar to the standard graphing calculator syntax used throughout by the system, with the addition of some new functions.

In this question, the randomized variables are mass, **$m**, which varies between 0.5 and 1.0 and velocity, **$v**, which varies between 10 and 15. The decimal argument is used to truncate the values of the numeric variable data.

The answer field holds two statements, one corresponding to each part of this multipart formula question. Statements in the *answer* field are evaluated by the system grader, so must be expressed using proper math syntax. Also, note that in this question we use the **comment** field to display the computed value of the answer rather than **answer** definition (Displaying Answers).

See Also:

Overview of Algorithmic Question Capabilities (page 217)

10.24 Tutorial: List-based Variables

You can use the algorithmic syntax to manage multiple scenarios or sets of randomized data in questions. Using list-based variables, you can use the same question statement but substitute corresponding sets of data throughout the question, answer, comment, hints, and solution statements.

Example

Consider the case in which you want to create a chemistry problem that uses one of several sets or scenarios of data related to distinct elements. Each set contains a visual depiction (**GIF** file) and related data linked to various properties of the distinct elements.

To create a single problem that randomly chooses an element and substitutes its individualized properties into the question statement and other fields, use algorithm and answer statements as in the following examples.

(a) Single Index Case with Sixteen Scenarios

1. Use a single variable **$k** as an index.

```
$k=rint(15);
```

2. Create individual variables for the first related property of each of the sixteen scenarios. In this case, the first related property is an image file depicting the molecular structure. Create a new variable $molecule1 that uses the index to select the appropriate indexed data value (in this case, the correct image).

```
$molecule1x00="<img src=../classes/myclass/chap15/image01.gif>";
$molecule1x01="<img src=../classes/myclass/chap15/image02.gif>";
$molecule1x02="<img src=../classes/myclass/chap15/image03.gif>";
...
$molecule1x15="<img src=../classes/myclass/chap15/image16.gif>";
$molecule1=switch($k, $molecule1x00, $molecule1x01,  $molecule1x02, ... ,$molecule1x15);
```

3. Define the second related property for each scenario. Create a new variable that uses the index to select the appropriate indexed data value.

```
$property1x00="value...";
$property1x01="value...";
...
$property1x15="value...";
$property1=switch($k, $property1x00, $property1x01,  ... , $property1x15);
```

4. Define additional related properties for each scenario, and then select (using the index) and assign the appropriate values to new variables.

5. The final set of variables is the answer field (assuming that each different case has a different answer). Select the appropriate answer using the index.

```
$ans00=50;
$ans01=16;
..
$ans15=3.6;
$ans=switch($k, $ans2x00, .. , $ans2x15);
```

(b) Multiple Index Case

It may be the case that your question requires multiple indexes to your data. Consider the above case with only four elements (**molecule1** to **molecule4**), where for each element there are four individualized sets of properties.

Assume that you have four images of molecules. To avoid redundancy (because **molecule1** has only four values, but there are sixteen scenarios), use two indices. The first index, **$i**, determines the element and image. The second index, **$k**, determines the scenario. Depending on the value of **$i**, the value of **$k** is in one of the four ranges: 0-3, 4-7, 8-11, or 12-15. By using **$k**, **$property1** (and other properties, and **$ans**) match with the appropriate **molecule1.** To implement this, use the following statements.

```
$i=rint(4);
$j=rint(4);
$k=4*i+j;

$molecule1x0="...";
$molecule1x1="...";
$molecule1x2="...";
$molecule1x3="...";
$molecule1=switch($i, $molecule1x0, $molecule1x1,..);

$property1x00="..";
$property1x01="..";
..
$property1x15="..";
$property1=switch($k, $property1x00, .. , $property1x15);

$ans00=50;
$ans01=16;
..
$ans15=3.6;
$ans=switch($k, $ans2x00, .. , $ans2x15);
```

See Also:

switch(n, a, b, c, ...) (page 232)

10.25 Tutorial: Maple-based Algorithmic Question Authoring

Algorithmically generated variables are used to generate random numbers in questions and create multiple permutations of questions from a single template.

You can use either Maple T.A. or Maple to generate algorithmic variables. You can use either (or both) types of variables in *any* question type.

The syntax for a Maple-based variable definition is:

```
<variable_i>=maple("<Maple_command>");
```

Using a Text or HTML Editor

To define an algorithmic variable in the plain-text script file, specify variable definitions (or other variable control sequences) in the **algorithm** field.

```
algorithm=
<variable_definition_1>
...
<variable_definition_n>
@
```

Using the Question Editor

In the **Question Editor**, define variables by directly entering Maple-based variable definitions on the algorithm screen or using the Algorithm Designer.

Tutorial 1

The following tutorial shows how to convert a static **Maple Formula** question to an algorithmically generated question by using variables assigned values by Maple commands.

Maple-graded Question

Consider the following plain-text script question definition, which asks a student to enter an anti-derivative of **x^3**.

```
qu.1.1.question=Enter an anti-derivative of x^3.@
qu.1.1.maple=evalb(diff($RESPONSE,x)=x^3)@
qu.1.1.type=formula@
qu.1.1.mode=Maple@
qu.1.1.allow2d=1@
qu.1.1.name=Maple AntiDerivative@
qu.1.1.comment=Any function of the form x^4/4 + C, where C is a constant, is an
anti-derivative of x^3.@
qu.1.1.editing=useHTML@
```

Notes:

1. This question uses Maple to verify that the derivative of the student response is equal to **x^3**. This is recommended in place of comparing the student response, which can have an additive constant, with the integral of **x^3**, that is evalb($RESPONSE=int(x^3,x)).
2. If the student enters an incorrect response, the system display the value in the **comment** field. In Maple-graded questions, if there is no **comment** field, the system returns the message **Comment: No feedback provided with this question**.

To change this question to an algorithmic question, use the following plain-text script question definition.

```
qu.1.2.question=Enter an anti-derivative of x^$exponent + ${coeff}x.@
qu.1.2.maple=evalb(diff($RESPONSE,x)=x^$exponent+$coeff*x)@
qu.1.2.type=formula@
qu.1.2.mode=Maple@
qu.1.1.allow2d=1@
qu.1.2.name=Maple AntiDerivative@
qu.1.2.comment=Any function of the form $answer + C, where C is a constant, is an
anti-derivative of x^$exponent + ${coeff}x.@
qu.1.2.editing=useHTML@
qu.1.2.algorithm=$exponent=maple("randomize():rand(2..5)()");
$coeff=maple("randomize():rand(2..9)()");
$answer=maple("int(x^$exponent+$coeff*x,x)");
@
```

Notes:

1. You must follow the variable naming conventions.
2. For a list of the guidelines for Maple commands in Maple-based variables, see *The Algorithm Designer (page 80)*.
3. The variable **$exponent** is assigned a value calculated by Maple using the **rand** function.
   rand() returns a random **12**-digit positive integer
   rand(a)() returns a random integer in the range **0 ... a-1** (inclusive)
   rand(a..b)() returns a random integer in the range **a ... b** (inclusive)
   For information on other Maple commands, see your Maple documentation.
4. The Maple command:
   randomize():
   is included so that a different value is generated for each question instantiation. You must include the **randomize():** command in every Maple-based variable definition. That is, it is required in the definition of **$exponent** and **$coeff**.
5. The Maple command:
   int(x^$exponent,x)
   calculates the correct answer, which is displayed using the **comment** statement if the student response is incorrect.
6. The examples in **Tutorial 1** can be converted to **Maple Syntax** questions by changing the type field value from **formula** to **maple**, and changing allow2d to 0 for text-entry mode. For **Maple Syntax** questions, it is recommended that you attempt to prevent a student from calculating the correct answer using Maple commands. For more information, see *Using Maple Code to Prevent Cheating in Maple Syntax Questions (page 185)*.

For an example plain-text script that defines a Maple question with Maple plotting of the student response, see *Plotting a Student Response (page 187)*.

Tutorial 2

You can also use Maple-based algorithmic variables in other question types.

Formula Question

The following plain-text script defines a **Formula** question that uses Maple-based variables.

```
qu.1.3.question=What is the sum of $a and $b?@
qu.1.3.mode=Formula@
qu.1.3.name=Maple-based Variables in Formula@
qu.1.3.answer=$answer@
qu.1.3.editing=useHTML@
qu.1.3.algorithm=$a=maple("randomize():rand(100..200)()");
$b=rint(10,50);
$answer=int($a+$b);
@
```

Notes:

1. This question uses both Maple-based (for **$a**) and Maple T.A. random variables (for **$b**).
2. When referencing negative, random variables in a **maple** variable definition, be sure to place the negative variable in parentheses, otherwise an error message will return. **$c=maple("$a + ($b)");**

See Also:

Maple-graded Formula Question Example Script (page 293)

Maple-graded Question Type - Overview (page 140)

Maple Syntax Question Example Script (page 294)

Questions with Plots (page 148)

11 Authoring Questions in LaTeX

11.1 LaTeX Authoring Overview

You can author and edit *all* question types using LaTeX to generate plain-text script files.

There are certain advanced question types that *cannot* be created in the Question Editor (for example, dynamically rendered plots in questions, algorithmically generated labels in figures, matrices, and dynamic graph sketching). For these questions, you can use plain text script files or LaTeX.

It is recommended that you write and edit question banks using LaTeX only if you are an experienced LaTeX user. Otherwise, it is recommended that you edit script files using a text or HTML editor or use the Question Editor.

Requirements

To use LaTeX to author **Maple T.A.** question banks, you need:

- **ed.sty** - The **Maple T.A.** LaTeX macros. You can download this file at: http://www.maplesoft.com/products/mapleta/latex2ta/ by choosing the default category.
- A text editor
- LaTeX installed on your system
- Internet access

Authoring Process

To author in LaTeX:

1. Write your question bank in LaTeX using a text editor.
2. Run LaTeX on your text files. Preview the output to ensure it is correct.
   You may perform multiple iterations of the steps 1 and 2.
3. Using the web-based conversion service at http://www.maplesoft.com/products/mapleta/latex2ta/, convert your LaTeX-format question file to Maple T.A. **.qu** format. Save the .**qu** file when the conversion is complete.
4. Import and save the converted question bank in your Maple T.A. class. (See *Opening a Saved Question Bank File (page 120)*.)

11.2 LaTeX Question Bank Document Structure

Questions are grouped into topics. Use the LaTeX **topic** environment to mark the beginning and end of topics.

Each question is wrapped inside a **question** environment. The Maple T.A. question type must be provided as an argument to the opening statement.

You can include any number of questions. The example format in **Table 11.1** shows one topic containing two questions.

The LaTeX package **epsfig** is loaded by default. The **ed** package is required. Other standard LaTeX packages, for example, **amsmath**, can be included using the **\usepackage** macro. You can also define additional LaTeX macros as usual.

To include graphics:

1. Select a location in your Maple T.A. class web site for graphics files. It is recommended that you use a distinct folder for the images in each question bank. Upload the graphic to your class web site. For information on uploading, see *Uploading Images and Reference Files to the Class Web Site (page 87)*.
2. Insert the **\setImageBase** directive near the top of your question bank TeX source file to specify the path to this location. For example, if your Maple T.A. class name is *calculus205*, and your graphics folder is the *chapter1* folder of your Maple T.A. class web site, use the directive:

```
\setImageBase{../classes/calculus205/chapter1/}
```

Note: The final slash is optional.

Table 11.1: Example LaTeX Document Structure

```
\documentclass[12pt]{article}
\usepackage{ed}
% Optionally set the location for graphics files on the Maple T.A. system.
\setImageBase{../classes/Maple T.A.classname/images_folder/}
\begin{document}
\begin{topic}{...}
\begin{question}{question type}
...
\end{question}
\begin{question}{question type}
...
\end{question}
\end{topic}
\end{document}
```

11.3 Question Structure

Table 11.2 gives the basic structure for a question.

Table 11.2: Basic LaTeX Question Structure

```
\begin{question}{question type}
\name{question name}
\property{name1}{value1}
\property{name2}{value1}
\hint{first hint}
\hint{second hint}
\comment{feedback text}
\qutext{question text}
...
\end{question}
```

Available Fields in LaTeX Questions

All questions can be given a name, hints, a comment, and a set of properties.

- The **\name** field is used as an identifier for a question. It is recommended that you assign a **name** to every question. This will help you to identify questions when creating assignments from your question bank.
- If present, the **\comment** field provides feedback that can be shown to the student instead of the **\answer** field if the question is answered incorrectly.
- The **\hint** field provides hints for the question. There can be more than one **\hint** statement.

- The **\property** field takes two arguments. The first is the name of the property. The second is its value. You can choose any useful name. There can be more than one **\property** statement. When converted to the Maple T.A. format files, properties are converted to **information** fields.

Example

```
\begin{question}{formula}
\name{p65, qu14}
\qutext{Differentiate $x e^x$.}
\answer{x e^x + e^x}
\comment{Use the product rule to differentiate this formula. The answer
is $x e^x + e^x=(x+1)e^x$.}
\hint{Use the product rule.}
\property{author}{I. M. Prof}
\property{difficulty}{medium}

% The 'author' and 'difficulty' property names are
% examples. You can create property names as required.
\end{question}
```

Other macros include:

- **\code** - To specify algorithmic statements, use the **\code** command. For example:
  \code{
  $m=rint(7);
  $n=int(2*$m+3);
  }
- **\var** - To use a system algorithmic variable in LaTeX, do *not* specify the **$**, but instead enclose the name in a **\var{}** command. For example, to display the variable **$n** in the question statement, use:
  \qutext{What is the exact value of $\sin(\frac{\var{n}\pi}{4})$?}
  For more information, see *Tutorial: Using Algorithms in LaTeX Questions (page 264)*.

In addition, each question type has its own set of LaTeX macros for defining properties of that type of question.

See Also:

Overview of Algorithmic Question Capabilities (page 217)

The info Field (page 128)

11.4 LaTeX Style Sheets

The Maple T.A. LaTeX macros can be used on a range of platforms and with a variety of TeX implementations. To determine where to save your style sheets (**.sty** files), refer to your LaTeX documentation.

11.5 Basic Question Types in LaTeX

Blanks Questions in LaTeX

Question Mode: Blanks - [menu] option

The **Blanks** question type generates a fill-in-the-blank question. The text inside a **\blank[menu]** command is removed and a list of choices is displayed in a drop-down box.

Notes

- You must provide the choices in a comma-delimited list, with the correct answer listed first. The choices are permuted for the student.
- This option is similar to *Multiple Choice*, but may be more convenient for some questions. The main difference is that only text choices are supported. TeX cannot be used in the list of choices.
- Another advantage of the **Blanks** question type is that several **\blank** macros can be included in a single question and different **\blank** options can be mixed in a single question. This makes the **Blanks** type powerful and flexible.
- Blanks must not appear in math mode.
- A limited amount of error recovery exists.

  For example,

  ```
  '$x = \blank[menu]{0,1,2}$'
  ```

  converts correctly, but

  ```
  '$$x = \blank[menu]{0,1,2}$$'
  ```

  and

  ```
  '$\int_0^{\blank[menu]{1,2}} f(x)\,dx$'
  ```

  generate conversion errors.

Example 1

```
\begin{question}{Blanks}
\qutext{The
% Format: \blank[menu]{correct choice, other choices}
\blank[menu]{human, canada goose, garter snake, housefly} and
\blank[menu]{chimpanzee, emu, frog, electric blue hap} are mammals.}
\end{question}
```

Example 2

```
\begin{question}{Blanks}
% This question mixes the [menu] and [formula] options.
\qutext{The graph of the function $f(x)=(x-3)^2$ is the same as the
graph of $g(x)=x^2$ shifted to the \blank[menu]{right, left}
\blank[formula]{3} units.}
\end{question}
```

Example 3

```
\begin{question}{Blanks}
% This question uses randomized variables.
\qutext{The graph of the function
$f(x)=\var{s1}\var{a}(x+\var{h})^2-\var{k}$ is the same as the graph of
```

```
the function $g(x)=\var{s1}\var{a}x^2$ shifted
\blank[menu]{to the left \var{h} units,to the right \var{h} units,
up \var{h} units,down \var{h} units}
and shifted \blank[menu]{down \var{k} units,up \var{k} units,
to the right \var{k} units, to the left \var{k} units}.}
\code{
$t1=rint(2);
$s1=if($t1,"","-");
$a=int(rint(8)+2);
$h=int(rint(8)+2);
$k=int(rint(8)+2);
}
% The above code selects random integer coefficients between 2 and 9,
% and allows the leading coefficient of the function f to be either
% positive or negative.
\end{question}
```

Example 4

```
\begin{question}{blanks}
% This question uses randomized variables in the menu choices and
% other parts of the question.
\qutext{In the \var{q} quadrant, the graph of $y=\var{c}(\var{a1})^x$ is
\blank[menu]{\var{p1},\var{p2}} the graph of $y=\var{c}(\var{a2})^x$.}
\code{
$r=rint(2);
$s=rint(2);
$t=int(rint(8)+2);
$q=switch($r,"first","second");
$p1=switch($r+$s,"above","below","above");
$p2=switch($r+$s,"below","above","below");
$c=int(rand(2,30));
$a=rand(1.1,12.1,2);
$a1=switch($s,$a+$t,$a);
$a2=switch($s,$a,$a+$t);
}
\end{question}
```

See Also:

Free Response and Fill-in-the-Blank Questions (page 156)

Clickable Image Questions in LaTeX

Question Mode: Clickable Image

The **Clickable Image** question type displays an image and text. As the student moves the mouse over the image, various hot-spots are selected. The goal is to click the correct hot-spot.

To write a clickable image question you must first upload the image:

1. Select a location in your Maple T.A. class web site for graphics files. It is recommended that you use a distinct folder for the images in each question bank. Upload the graphic to your class web site. For information on uploading, see *Uploading Images and Reference Files to the Class Web Site (page 87)*.
2. Insert the **\setImageBase** directive near the top of your question bank TeX source file to specify the path to this location. For example, if your Maple T.A. classname is *calculus205*, and your graphics folder is the *chapter1* folder of your Maple T.A. class web site, use the directive:

```
\setImageBase{../classes/calculus205/chapter1/}
```

Note: The final slash is optional.

To incorporate your image file into the question, use the **clickableimage** environment. The **\begin{clickableimage}** command takes three arguments:

(i) The name of the image file.

(ii) The width of the image file in pixels.

(iii) The height of the image file in pixels.

To determine the width and height of the graphic, use a third-party graphics application (for example, Microsoft® Paint).

To define the hotspots, use the **\region** macro. The argument to the **\region** macro is a comma-delimited list of points, specifying the vertices of the clickable region.

- The x and y-coordinates are measured from the upper left corner of the graphic.
- More than one **\region** command can be used. Mark the correct region with an asterisk.
- Note that **\begin{clickableimage}...\end{clickableimage}** must appear outside the argument to **\qutext**.

Note: The **\setImageBase** directive has a global effect on your document. You can temporarily override the image base with **\ignoreImageBase**. This macro takes no arguments. Its effect is local, so upon leaving the current group or environment, the image base will revert to its previous value.

Example

```
\begin{question}{Clickable Image}
\qutext{Find the dog in the picture below.}
{\ignoreImageBase % use empty image base in this group
\begin{clickableimage}
  {http://www.YourDomain.com/images/pose.gif}{192}{288}
  \region*{139,32,   144,40,   137,58,   115,68,   109,101,   132,155,
           138,209,   155,273,   148,281,   52,282,   44,266,   29,224,
           33,198,   36,176,   61,108,   59,57,   74,37,   93,29}
  \region{149,87,   168,90,   172,148,   151,122}
  \region{153,157,   186,152,   185,212,   158,218,   149,185}
\end{clickableimage}} % image base returns to original value
\end{question}
```

Free Response Text-based Questions in LaTeX

The system supports three varieties of *open* text response questions:

- **Key Words**
- **Essay** (ungraded)
- **Blanks[text]**

Question Mode: Key Words

The **key words** question type is an open response question. The answer field consists of a segment of text in which the keywords are marked by enclosing parentheses. Any student response that includes all the words in parentheses is graded correct. If the question is not answered correctly, the full answer text (excluding the parentheses) is displayed as the correct answer.

Example

```
\begin{question}{key words}
\qutext{Which two scientists independently discovered inductance?}
\answer{Michael (Faraday) and Joseph (Henry)}
% Any responses that include both the words "Faraday" and "Henry"  is
% graded correct. If the question is not answered correctly,
% "Michael Faraday and Joseph Henry" is displayed as the correct answer.
\end{question}
```

Question Mode: Essay

The **essay** question type is an open response question. The system does not grade it. You must grade each student's answer and assign a score using the Gradebook.

Example

```
\begin{question}{essay}
\qutext{Discuss the effect of gamma rays on Man-in-the-Moon Marigolds.}
\end{question}
```

Question Mode: Blanks - [text] option

The **blank[text]** question type creates a fill-in-the-blank question. The text inside a **\blank[text]** command is removed and a drop-down box is displayed.

Notes

- You must provide the correct answer as an argument to the macro.
- The **[text]** option is the default. That is, **\blank{}** and **\blank[text]{}** are equivalent.
- An advantage of the **Blanks** question type is that several **\blank** macros can be included in a single question and different **\blank** options can be mixed in a single question. This makes the **Blanks** type powerful and flexible.
- **Blanks** *cannot contain TeX math-mode content.* Blanks must not appear in math mode. Only raw text can be used in blanks.

Example

```
\begin{question}{Blanks}
\qutext{The human race belongs to the genus
% Format: \blank{correct answer}
\blank{Homo},
```

```
and species \blank{sapiens}.
}
\end{question}
```

See Also:

Free Response and Fill-in-the-Blank Questions (page 156)

Ungraded Essay Questions (page 155)

Matching Questions in LaTeX

Question Mode: Matching

The **Matching** question type presents two lists. The student must match each entry on the first list with one entry from the second list.

Notes

- To specify pairs of matching elements, use the **\match** (for elements in the first list) and **\with** (for elements in the second list) macros.
- You can present additional entries on the second list that do not match any items on the first list. Use the **\decoy** macro for this purpose.
- The default behavior is to display the first list in a single row across the screen, with a drop-down list of all choices from the second list. You can instead specify the number of columns using the **\cols** macro.

Example 1

```
\begin{question}{Matching}
\qutext{Match the following states with their capitals.}
\match{California} \with{Sacramento}
\match{New York} \with{Albany}
\match{Pennsylvania} \with{Harrisburg}
\match{Oregon} \with{Salem}
\end{question}
```

Example 2

```
\begin{question}{Matching}
\qutext{Match the following polynomials with their factorizations:}
\match{$x^2-1$} \with{$(x-1)(x+1)$}
\match{$x^2+2x+1$} \with{$(x+1)^2$}
\match{$x^2-2x+1$} \with{$(x-1)^2$}
% The optional \decoy terms are presented as extra choices. Take care
% that they are not possible alternate correct answers!
\decoy{$(x+1)(x+2)$}
\decoy{$(x+3)(x+2)$}
% Optionally, specify the number of columns in which to arrange the display of data.
\cols{3}
\end{question}
```

See Also:

Matching Questions (page 161)

Multipart Questions in LaTeX

Question Mode: Multipart

To make a multipart question, declare a question of type **Multipart** and include other questions in its body.

Notes

- **Multipart** questions can be nested inside other **Multipart** questions. The conversion tool accepts only four levels of nesting

```
\begin{question}{multipart}
...
\begin{question}{multipart}
...
\begin{question}{multipart}
...
\begin{question}{multipart}
...
\begin{question}{multipart}
...
\end{question}
\end{question}
\end{question}
\end{question}
\end{question}
```

- **Multipart** questions can contain any question types as parts.
- The parts of a question are numbered using lower-case letters regardless of nesting level. To use different labels, use the **numbering** macro. Its argument must be one of:
  alpha - "**(a)**, **(b)**, **(c)**, **(d)**, ..." (the default setting)
  Alpha - "**(A)**, **(B)**, **(C)**, **(D)**, ..."
  roman - "**(i)**, **(ii)**, **(iii)**, **(iv)**, ..."
  Roman - "**(I)**, **(II)**, **(III)**, **(IV)**, ..."
  arabic - "**(1)**, **(2)**, **(3)**, **(4)**, ..."
  none - no numbering
- By default, each part receives equal weighting when the question is graded. To specify a non-uniform weighting, use the **weighting** macro prior to the **begin{question}** for the first part. The **weighting** macro requires one argument, a comma-delimited list of non-negative integers, specifying the relative values assigned to each part.

Example

```
\begin{question}{Multipart}
\weighting{1,2} % The value of part (b) is twice that of part (a).
\qutext{Answer the following questions:}
% part (a)
\begin{question}{Formula}
\qutext{Differentiate the function
$\displaystyle f(x)=\frac{x}{1+x^2}$.}
\answer{(1 - x^2)/(1 + x^2)^2}
\end{question}
% part (b)
\begin{question}{Multipart}
% The 2/3 of the points assigned to part  (b) are split evenly between
% its subparts because \weighting is not used.
\numbering{roman} % These are subparts (i)  and (ii)
% part (i)
\begin{question}{Ntuple}
\qutext{Find the absolute minimum point on the graph of the
function $\displaystyle f(x)=\frac{x}{1+x^2}$.}
\answer{(-1,-1/2)}
\end{question}
```

```
% part (ii)
\begin{question}{Ntuple}
\qutext{Find the absolute maximum point on the graph of the
function $\displaystyle f(x)=\frac{x}{1+x^2}$.}
\answer{(1,1/2)}
\end{question}
\end{question}
\end{question}
```

See Also:

Multipart Questions (page 162)

Multiple Choice Questions in LaTeX

Question Mode: Multiple Choice; Nonpermuting Multiple Choice

The **Multiple Choice** and **Nonpermuting Multiple Choice** question types create a question and a list of choices, which includes one correct answer.

Notes

- You can have as many (or as few) choices as desired. Mark the correct choice with an asterisk.
- The order of the choices is permuted for the student. To force the choices to appear in the order you specify, use the **Nonpermuting Multiple Choice** question type.

Example 1

```
\begin{question}{MultipleChoice}
\qutext{What is the capital of California?}
% List the choices, and mark the correct choice with an asterisk.
\choice{San Francisco}
\choice{Los Angeles}
\choice*{Sacramento}
\end{question}
```

Example 2

```
\begin{question}{MultipleChoice}
\qutext{$3\log x-2\log y=$}
\choice*{$\log\left(\displaystyle\frac{x^3}{y^2}\right)$}
\choice{$\log(x^3y^2)$}
\choice{$\log(3x-2y)$}
\choice{$\log(x^3-y^2)$}
\end{question}
```

Example 3

```
\begin{question}{NonPermutingMultipleChoice}
% This question uses randomized variables.
\qutext{Consider the function $f(x)=\var{c}x^{\var{n}}$.
What happens to $f(x)$  as $x\to -\infty$?}
\choice*{$f(x)\to\infty$}
\choice{$f(x)\to  -\infty$}
\choice{$f(x)\to  0$}
\code{
$n=int(2*rint(4)+3);
$c=int(-(rint(6)+2));
}
% The above code chooses random integer coefficients n and c.
% The exponent n is positive and odd (3, 5, 7, or 9).
```

```
% The coefficient c is negative, between -2 and -7.
\end{question}
```

See Also:

Multiple Choice Questions (page 163)

Multiple Selection Questions in LaTeX

Question Mode: Multiple Selection; Nonpermuting Multiple Selection

The **Multiple Selection** and **Nonpermuting Multiple Selection** question types create a question and a list of choices, which includes possibly more than one correct answer.

Notes

- You can have as many (or as few) choices as desired. Mark the correct choice(s) with an asterisk.
- The order of the choices is permuted for the student. To force the choices to appear in the order you specify, use the **Nonpermuting Multiple Selection** question type.

Example 1

```
\begin{question}{Multiple Selection}
\qutext{Select all cities that are located in California.}
% List the choices, and mark all the correct choices with an asterisk.
\choice*{San Francisco}
\choice*{Los Angeles}
\choice{Chicago}
\choice{New York}
\choice*{Sacramento}
\end{question}
```

Example 2

```
\begin{question}{MultipleSelection}
\qutext{Select the equations of all lines that are vertical asymptotes
of the graph of $y=\sec x$. If the graph has no vertical asymptotes,
select ``No vertical asymptotes''.}
\choice*{$x=-7\pi/2$}
\choice{$x=-\pi$}
\choice*{$x=-\pi/2$}
\choice{$x=0$}
\choice{$x=\pi/4$}
\choice{$x=2\pi/3$}
\choice*{$x=5\pi/2$}
\choice{No vertical asymptotes}
\end{question}
```

Example 3

```
\begin{question}{MultipleSelection}
% This question uses randomized variables.
\qutext{Select the equations of all lines that are vertical asymptotes
of the graph of the function
$f(x)=\displaystyle\frac{\var{a}x^{\var{b}}}{x^2-\var{cc}}$.
If the graph has no vertical asymptotes, select ``No vertical
asymptotes''.}
\choice*{$x=-\var{c}$}
\choice*{$x=\var{c}$}
\choice{$x=0$}
\choice{$x=\var{d}$}
\choice{$x=\var{e}$}
\choice{No vertical asymptotes}
\code{
```

```
$a=int(rint(8)+2);
$b=int(rint(8)+2);
$c=int(rint(4)+1);
$cc=int($c^2);
$t=rint(2);
$d1=int($c+rint(2)+1);
$d=int(switch($t,$d1,-$d1));
$e1=int($d+rint(2)+1);
$e=int(switch($t,-$e1,$e1));
}
% The above code chooses random integer coefficients a and b between
% 2 and 9. The value of cc can be 1, 4, 9, or 16. d and e are
% extra choices that do not duplicate the others.
\end{question}
```

See Also:

Multiple Selection Questions (page 164)

True-or-False Questions in LaTeX

Question Mode: True False

The **True False** question type creates a question and a list of two choices: **True** and **False**.

Example 1

```
\begin{question}{True False}
\qutext{Whales are mammals.}
% Mark the answer \true or \false:
\true
\end{question}
```

Example 2

```
\begin{question}{TrueFalse}
\qutext{The equation $2x-3y^2=4$ defines a function with an independent
variable $x$.}
\false
\end{question}
```

Example 3

```
\begin{question}{TrueFalse}
% This question uses randomized variables.
\qutext{The equation $\var{a}x-\var{b}y^2=\var{c}$ defines a function
with an independent variable $x$.}
\false
\code{
$aa=int(rint(11)+2);
$s=rint(2);
$a=int(if($s,$aa,-$aa));
$b=int(rint(11)+2);
$cc=int(rint(11)+2);
$t=rint(2);
$c=int(if($t,$cc,-$cc));
}
% The above code chooses random integer coefficients with absolute values
% between 2 and 12. b must be positive, but a and c may be
% positive or negative.
\end{question}
```

See Also:

True-or-False Questions (page 177)

11.6 Advanced Question Types in LaTeX

Graph Plotting in LaTeX Questions

Graph Plotting Applet - \graph and \multigraph Macros

The **\graph** macro embeds a graph in a question. (You can also embed Maple plots in questions using the **Question Editor**.) The graph has its center at the origin, x and y-axes, and horizontal and vertical gridlines. The axes are labeled at each gridline.

The macro takes one option and five required arguments, in the following order:

Table 11.3: Arguments and Option for a Graph Plotting Applet

gridlines option	number of horizontal and vertical gridlines in addition to the axes (default is 10)
formula	formula of the function in calculator syntax, not TeX
xMin	minimum x-value on the x-axis
xMax	maximum x-value on the x-axis
yMin	minimum y-value on the y-axis
yMax	maximum y-value on the y-axis

The **\multigraph** macro plots two curves on the same axes. This macro takes one option and six required arguments, in the following order:

Table 11.4: Arguments and Option for a Multigraph

gridlines option	number of horizontal and vertical gridlines in addition to the axes (default is 10)
formula1	formula of first function in calculator syntax, not TeX
formula2	formula of second function in calculator syntax, not TeX
xMin	minimum x-value on the x-axis
xMax	maximum x-value on the x-axis
yMin	minimum y-value on the y-axis
yMax	maximum y-value on the y-axis

Syntax

```
\graph[gridlines]{formula}{xMin}{xMax}{yMin}{yMax}
```

\multigraph[gridlines]{formula1}{formula2}{xMin}{xMax}{yMin}{yMax}

Example

```
\begin{question}{MultiFormula}
\qutext{What are the $x$-intercepts of the graph shown?
% Format: \graph[gridlines]{formula}{xMin}{xMax}{yMin}{yMax}
\graph[12]{(x-1)(x+3)}{-6}{6}{-6}{6}}
% The choice of gridlines and scale here will ensure that the axes are
% labeled at the integers -6, -5, ..., -1, 1, ..., 5, 6.
\answer{1;-3}
\end{question}
```

See Also:

Questions with Plots (page 148)

Graph Sketching Questions in LaTeX

Question Mode: sketch

The **sketch** question type displays introductory text accompanied by a set of axes. The student draws a graph on the axes by clicking points to be used as interpolation nodes.

Use the **sketch** environment to configure the applet, declaring the ranges of the axes, a sample correct response, and a list of criteria that the student answer must meet. The **\begin{sketch}** macro takes one option and four required arguments, in the following order:

Table 11.5: Arguments and Option for a Sketch

gridlines option	number of horizontal and vertical gridlines in addition to the axes (default is 10)
xMin	minimum x-value on the x-axis
xMax	maximum x-value on the x-axis
yMin	minimum y-value on the y-axis
yMax	maximum y-value on the y-axis

The following commands must appear in the **sketch** environment.

- **\example{<point-list>}** describes a sample correct answer that is displayed if the student gives an incorrect response. The sketch is given as <point-list>, a space-delimited set of coordinates of the form **x,y**. The resulting sketch is a single curve interpolated from these points.
- **\check{criterion}** sets a criterion that the student's sketch must satisfy. Available criteria are:

Table 11.6: Basic Sketch Criteria

goes_through(xcoord, ycoord)	sketched function must pass through the point **(xcoord, ycoord)**
increasing	sketched function must be increasing
decreasing	sketched function must be decreasing
concave_up	sketched function must be concave up
concave_down	sketched function must be concave down
linear	sketched function must be linear

To restrict a criterion to an interval **[a,b]**, use **\check[a,b]{criterion}**. This option is ignored for **goes_through**.

The criterion can also be a boolean expression; useful statements in this case are:

Table 11.7: Statements for Additional Sketch Criteria

slope_at(xval)	return the slope of the curve at the x-value **xval**.
value(xval)	return the y-value for the x-value **xval.**
expr1 == expr2	return **true** if **expr1** is equal to **expr2**. Otherwise, it returns **false**.
expr1 < expr2	return **true** if **expr1** is less than **expr2**. Otherwise, it returns **false**.
expr1 > expr2	return **true** if **expr1** is greater than **expr2**. Otherwise, it returns **false**.
true	boolean constant
false	boolean constant
bool1 && bool2	return **true** if **bool1** and **bool2** evaluate to **true**. Otherwise, it returns **false**.

bool1 || bool2	return **true** if **bool1** or **bool2** (or both) evaluate to **true**. Otherwise, it returns **false**.

For example, **value(xcoord) == ycoord** is equivalent to **goes_through(xcoord, ycoord)**.

More than one **\check** statement can appear, in which case every criterion must be satisfied. That is,

```
\check{criterion1}
\check{criterion2}
\check{criterion3}
```

is equivalent to

```
\check{criterion1 && criterion2 &&  criterion3}
```

Syntax

```
\begin{sketch}[gridlines option]{xMin}{xMax}{yMin}{yMax}
  \example{<point-list>}
  \check{criterion1}
  \check{criterion2}
  \check{criterion3}
\end{sketch}
```

Example 1

```
\begin{question}{sketch}
\qutext{Sketch the graph of the function $y=x+1$.}
\begin{sketch}[4]{-2}{2}{-2}{2}
  \example{-2,-1   1,2}
  \check{linear}
  \check{goes_through(0,1)}
\end{sketch}
\end{question}
```

Example 2

This example uses randomized variables.

```
\begin{question}{sketch}
\code{$a = int(rand(2,6));
      $a2  = int($a*$a);
      $xmax  = int(2*$a);
      $x  = sqrt(2)*$a;}
\qutext{Sketch the graph of the function $y=x^2/\var{a2}$.}
\begin{sketch}[4]{-\var{xmax}}{\var{xmax}}{-4}{4}
  \example{-\var{x},2   -\var{a},1   0,0  \var{a},1   \var{x},2}
  \check{goes_through(0,0)  && goes_through(-\var{a},1)
        &&  goes_through(\var{a},1)}
  \check{slope_at(0)  == 0}
  \check[-\var{x},0]{decreasing}
  \check[0,\var{x}]{increasing}
  \check{concave_up}
\end{sketch}
\end{question}
```

Including Graphic References in LaTeX Questions

Including Graphics Using \image

You can insert an image in a LaTeX question. The **\image** macro is the recommended method for including graphics in a question. If the file is available locally as an Encapsulated Postscript (**eps** or **epsf**) file, it is included (**\image** calls **\epsfig** to include the graphic).

You can also embed Maple plots in questions using the **Question Editor**.

Important: Do not explicitly call **\epsfig**. Using **\epsfig** directly does not ensure that proper Maple T.A. code results.

To include graphics:

1. Select a location in your Maple T.A. class web site for graphics files. It is recommended that you use a distinct folder for the images in each question bank. Upload the graphic to your class web site. For information on uploading, see *Uploading Images and Reference Files to the Class Web Site (page 87)*.
2. Insert the **\setImageBase** directive near the top of your question bank TeX source file to specify the path to this location. For example, if your Maple T.A. classname is *calculus205*, and your graphics folder is the *chapter1* folder of your class Maple T.A. web site, use the directive:

   ```
   \setImageBase{../classes/calculus205/chapter1/}
   Note: The final slash is optional.
   ```

3. Include graphics in a question using the **\image** macro. The format for the **\image** macro is:

   ```
   \image[extension]{filename}
   ```

   The **extension** argument is optional. It specifies the extension to be appended to the filename (the second argument) when the question bank is converted to Maple T.A. format. The default value is **jpg**.

   If the **filename** argument is a filename with an extension, the first argument is ignored. If the **filename** argument is a filename without an extension, the **.eps** extension is used in preview mode, and **extension** (or the default **.jpg**) is used in production processing.

Sample 1

```
\image{graphic}
```

Places the file **graphic.eps** from the current directory in the **dvi** output in preview mode. In production processing, it produces an image tag pointing to **graphic.jpg** in the directory specified by **\setImageBase**.

Sample 2

```
\image[gif]{graphic}
```

Places the file **graphic.eps** from the current directory in the **dvi** output in preview mode. In production processing, it produces an image tag pointing to **graphic.gif** in the **\setImageBase** directory.

Sample 3

```
\image{graphic.pct}
```

Does not place the file **graphic.pct** in the **dvi** output. In production processing, it produces an image tag pointing to **graphic.pct** in the **\setImageBase** directory.

Notes

- While the **\epsfig** macro allows various options, for example, **width**, **\image** does not support any options. Encapsulated Postscript files appear at full size in the **dvi** file.

- The conversion tools do not provide graphic conversion services. You must use third-party tools to produce web image files.

Example

```
\begin{question}{True False}
\qutext{The following graph is the graph of a function.\newline
True or False?\newline
\image{graphic}}
\true
\end{question}
```

Labeling Graphics in LaTeX Questions

Labeled Graphics - labelgraphic Environment

The **labelgraphic** environment allows you to add text labels to graphics (**gif** and **jpg** files).

To add text labels to graphics in your LaTeX question:

1. Select a location in your Maple T.A. class web site for graphics files. It is recommended that you use a distinct folder for the images in each question bank. Upload the graphic to your class web site. For information on uploading, see *Uploading Images and Reference Files to the Class Web Site (page 87)*.
2. Insert the **\setImageBase** directive near the top of your question bank TeX source file to specify the path to this location. For example, if your Maple T.A. classname is *calculus205*, and your graphics folder is the *chapter1* folder of your Maple T.A. class web site, use the directive:

   ```
   \setImageBase{../classes/calculus205/chapter1/}
   ```

Note: The final slash is optional.

3. To incorporate the image file into a question, use the **labelgraphic** environment. The **\begin{labelgraphic}** command takes three arguments:

   (i) The name of the image file.

   (ii) The width of the image file in pixels.

   (iii) The height of the image file in pixels.

   To determine the width and height of the graphic, use a third-party graphics application (for example, Microsoft Paint).

- To label the graphic, use the **\grlabel** macro. The **\grlabel** macro takes three arguments:

  (i) The text of the label.

  (ii) The x-coordinate of the label, in pixels.

  (iii) The y-coordinate of the label, in pixels.

  - The x and y-coordinates are measured from the upper left corner of the graphic.
  - More than one **\grlabel** command can be used.

Note: Without any **\grlabel** commands, the **labelgraphic** environment only incorporates the image file into a question. To include a graphic, it is recommended that you use the **\image** macro. For more information on the image macro, see *Including Graphic References in LaTeX Questions (page 262)*.

Example

```
\begin{question}{Numeric}
\qutext{A ball is thrown upward from the top of a 40 foot high tower at
an initial velocity of 64 feet/second. The graph of the height of the
ball versus time is shown below. What is the maximum height of the ball in feet? Express your
answer to an accuracy of at least 3 significant figures.
\begin{labelgraphic}{ball.gif}{250}{200}
  \grlabel{time (seconds)}{120}{10}
  \grlabel{height (meters)}{10}{100}
\end{labelgraphic}}
\answer{104 ? 1}
\end{question}
```

11.7 Tutorial: Using Algorithms in LaTeX Questions

Algorithmic Questions

Randomized variables can be incorporated into any question. Rules and recommendations for effective use of randomized variables are given below, illustrated by related examples.

- The code that creates and manipulates the randomized variables is contained in the **\code{}** section of the question. Each variable in the code section is designated with a **$** symbol, such as **$m**, **$ans**, and **$b2**. The first character after the **$** symbol must be a letter. Each code statement is of the form

```
$variable=expression;
```

  Separate multiple variable definitions with a semicolon.

> **Note:** TeX comments (**%**) cannot appear in the argument to **\code**. To include comments, use the **comment** control sequence.

  The usual arithmetic operations and functions can be used in the *expression* portion of the code statement. However, unlike most other programming languages, variables cannot be redefined in terms of themselves (for example, the statement **$a=$a+1;** is not allowed). For a description of the additional functions that can be used in the code sections, see *Functions within Algorithms (page 219)*.

- To use the variables in other sections of the question, enclose them in the **\var** macro. For example, to refer to the variable **$a**, use:

```
\var{a}
```

In the following examples, a **formula** question is designed through a series of refinements.

Example 1

```
\begin{question}{Formula}
\qutext{Find the solution of the linear equation
$\var{a}x+\var{b}=\var{c}$.}
\answer{(\var{c}-\var{b})/\var{a}}
\code{
$a=rint(12)+1;
$b=rint(12)+1;
$c=rint(12)+1;
}
\end{question}
```

The above code randomly determines integer coefficients **$a**, **$b**, and **$c** between **1** and **12**. The answer is calculated in the **\answer** section.

Although the above example does not cause errors, there are several side effects:

- The answer is displayed in the form **(7-3)/5** (for example). This provides an indication of the method for obtaining the answer, which may be inappropriate.
- The above format can return answers of the form **(7-7)/5**, which is more complicated than the simplified form, **0**.

To avoid these issues, rewrite this example as follows, using a new variable **$ans** (so that you return only the final answer).

Example 2

Refinement of **Example 1**.

```
\begin{question}{Formula}
\qutext{Find the solution of the linear equation
$\var{a}x+\var{b}=\var{c}$.}
\answer{\var{ans}}
\code{
$a=rint(12)+1;
$b=rint(12)+1;
$c=rint(12)+1;
$ans=($c-$b)/$a;
}
\end{question}
```

In this code, there are variable type issues.

- All variables are floating-point variables unless they are wrapped in an **int()** or **rint()** statement to make them integers (for example, **$m=int(14*5);**), or in double quotes **""** to make them strings (for example, **$s="north";**).

The statement **$a=rint(11);** creates an integer variable, but the line **$a=rint(11)+1;** creates a floating-point variable as a result of the **+1** operation. The same is true for the definitions of **$b** and **$c**. Using the above question, equations in the form **2.0x+9.0=4.0**, which is correct, but not as simple as possible. Change the code so that **$a**, **$b**, and **$c** are integers.

Example 3

Further refinement of **Example 2**.

```
\begin{question}{Formula}
\qutext{Find the solution of the linear equation
$\var{a}x+\var{b}=\var{c}$.}
\answer{\var{ans}}
\code{
$a=int(rint(12)+1);
$b=int(rint(12)+1);
$c=int(rint(12)+1);
$ans=($c-$b)/$a;
}
\end{question}
```

There is one more problem. The variable **$a** can equal **1**, in which case the equation has the form **1x+5=8**. To avoid this situation, impose the condition that **$a** be an integer between **2** and **12**.

Although **$c** can be **0** or negative, if **$b** is **0** or negative the equation is again unsimplified. Generalize the code to allow **$c** to be any integer between **-12** and **12**.

Example 4

```
\begin{question}{Formula}
\qutext{Find the solution of the linear equation
```

```
$\var{a}x+\var{b}=\var{c}$.}
\answer{\var{ans}}
\code{
$a=int(rint(11)+2);
$b=int(rint(12)+1);
$c=int(rint(25)-12);
$ans=($c-$b)/$a;
}
\end{question}
```

There is another minor problem. The answer **$ans** is a floating-point number. The system grades the question properly, but the displayed feedback may not have the correct form. If the student gives an incorrect response, the correct answer is displayed as a (potentially inexact) floating-point number, for example, **.666666667**, instead of **2/3**. To solve the problem, use a string variable to display the answer as a fraction.

Example 5

```
\begin{question}{Formula}
\qutext{Find the solution of the linear equation
$\var{a}x+\var{b}=\var{c}$.}
\answer{\var{ans}}
\code{
$a=int(rint(11)+2);
$b=int(rint(12)+1);
$c=int(rint(25)-12);
$num=int($c-$b);
$ans="$num/$a";
}
\end{question}
```

Note: Writing **$ans="($c-$b)/$a";** displays answers of the form **(11-6)/3**. First, calculate the numerator and store it as an integer in **$num**, and then create the fraction with the string **"$num/$a"**.

This last change introduces a new problem. The system displays correct answers of the form **4/2** or **3/3**. To make integer answers display as integers, implement one more change.

Example 6

Final refinement of the question.

```
\begin{question}{Formula}
\qutext{Find the solution of the linear equation
$\var{a}x+\var{b}=\var{c}$.}
\answer{\var{ans}}
\code{
$a=int(rint(11)+2);
$b=int(rint(12)+1);
$c=int(rint(25)-12);
$num=int($c-$b);
$ansint=int($num/$a);
$ans=if(int(eq($num/$a,$ansint)),"$ansint","$num/$a");
}
\end{question}
```

By using the intermediate variable **$ansint** and the string **"$ansint"** in the last statement (instead of directly using the string **"int($num/$a)"**, which produces answers of the form **"int(6/3)"**), integers display properly.

This final version allows some non-reduced fractions, for example, **2/4**, to appear as answers, but, in general, both the problem and the answer appear in a simple, standard form.

Example 7 - Question with a Plot

```
\begin{question}{Formula}
\qutext{Enter the question of the following plot.<br>
```

```
$\var{plot}$.}
\answer{x^2+\var{a}x+\var{b}}
\code{
$a=rint(2, 9);
$b=rint(0, 9);
$plot=plotmaple("plot(x^2+$a*x+$b, x=-10..10)");
}
\end{question}
```

To add a plot to a question in the question text, feedback, or hints, add an algorithmic variable using the **plotmaple** command. For more information, see *plotmaple(text) (page 227)*.

Use the following guidelines for algorithms:

- Ensure that all possible values of randomized variables are valid in the question text and the answer, and that the response will grade correctly.
- If you are using integers in the question text or answer, ensure that they are formatted properly.
- Define randomized variables so that you avoid expressions like **1x**, **0x**, **x+-3**, or **-3/-4**. If these values are required, write the question so that they display properly.
- Write the code so that if the student answers incorrectly, the correct answer appears in a standard form. In general, you must use strings, as shown in the final example.

11.8 Math Question Authoring using LaTeX

The LaTeX authoring method is ideal for creating sophisticated math question content. This method supports the following math question types.

- Formula (*Formula Questions in LaTeX (page 271)*)
- Blanks - **[formula]** option (*Mathematical Fill in the Blanks - [formula] option (page 268)*)
- Numeric (*Numeric Questions in LaTeX (page 276)*)
- Maple-graded (*Maple-graded Questions in LaTeX (page 272)*)
- Formula List (*Formula List Questions in LaTeX (page 271)*)
- Matrix (*Matrix Questions in LaTeX (page 274)*)
- Dimensioned Formula (*Maple Formula (page 273)*)
- Restricted Formula (*Restricted Formula Questions in LaTeX (page 279)*)
- Formula mod C (*Constants of Integration Questions in LaTeX (page 269)*)
- Equation (*Equation Questions in LaTeX (page 270)*)
- Ntuple (*Coordinates of Points and Vectors Questions in LaTeX (page 280)*)
- MultiFormula (*Multi Formula Questions in LaTeX (page 274)*)
- Multipart Formula (*Maple Formula (page 273)*)

Other Math Question Features

- Graph plotting applet - \graph and \multigraph macros
- Including graphics - \image macro
- Labeled graphics - labelgraphic environment

See Also:

Overview: Authoring Methods for Question Banks (page 73)

Math Question Types Comparison Table (page 133)

Mathematical Fill in the Blanks - [formula] option

Question Mode: Blanks - [formula] option

The **Blanks blank[formula]** question type generates a fill-in-the-blank question in which the answer is a formula. The text inside a **\blank[formula]** command is removed and a text field is displayed.

Notes

- You must provide the correct answer as an argument to the macro. The answer must be written in calculator syntax, not in TeX. Enter the answer as you expect the student to enter it (although equivalent answers are graded correct).
- This option is similar to the **Formula** question type, but is more convenient for some questions.
- An advantage of the **Blanks** question type is that several **\blank** macros can be included in a single question, and different **\blank** options can be mixed in a single question. This makes the **Blanks** type powerful and flexible.
- You can use **blank[formula]** questions to create a matrix question by setting the blank response cells in an HTML table.
- If the answer is a number (that is, does not have any variables), it is recommended that you specify in the question text whether an exact response or an approximation is required. In the case of an approximation, the margin of error must be specified in your correct answer. This can be done by:
  - Specifying a margin of error using the **?** operator

  For example, if the answer field were

  ```
  {2.34  ? 0.01}
  ```

  any response between **2.33** and **2.35** (inclusive) would be graded correct. In other words, the number following the **?** operator is the margin of error. For more information on tolerances, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.
- As noted above, blanks cannot contain TeX math-mode content. Blanks must not appear in math mode. Only raw text can be used in blanks. A limited amount of error recovery exists. For example,

  ```
  '$$x = \blank[formula]{sin(1)}$$'
  ```

  and

  ```
  '$\int_0^{\blank[formula]{sin(1)}}  f(x)\,dx$'
  ```

  generate conversion errors, but

  ```
  '$x  = \blank[formula]{sin(1)}$'
  ```

  converts correctly. However, it is recommended that you use a construction similar to:

  ```
  '$x  = {}$\blank[formula]{sin(1)}$'
  ```

Example 1

```
\begin{question}{Blanks}
\qutext{The derivative of the function $f(x)=4x-x^3$ is
% Format: \blank[formula]{correct answer in calculator syntax}
\blank[formula]{4-3x^2}
and the second derivative is \blank[formula]{-6x}.}
\end{question}
```

Example 2

```
\begin{question}{Blanks}
\qutext{The function $1-x^2$ is positive for values of $x$ in the range
```

```
\blank[formula]{-1} $< x <$ \blank[formula]{1}.}
\end{question}
```

Example 3

```
\begin{question}{Blanks}
\qutext{The maximum value of the function
$f(x)=\displaystyle\frac{x}{1+2x^2}$ is
\blank[formula]{0.353553 ? 0.0001} and the minimum value is
\blank[formula]{-0.353553 ? 0.0001}. Your answers must be correct to at least 4 decimal places.}
\end{question}
```

Example 4

```
\begin{question}{Blanks}
% This question mixes the [menu] and [formula] options.
\qutext{The graph of the function $f(x)=(x-3)^2$ is the same as the
graph of $g(x)=x^2$ shifted to the \blank[menu]{right, left}
\blank[formula]{3} units.}
\end{question}
```

Example 5

```
\begin{question}{blanks}
% This question uses randomized variables.
\qutext{An earthquake that measures $\var{a}$ on the Richter scale is
\blank[formula]{\var{ans}} times more powerful than one that measures
$\var{b}$. Your answer must be exact.}
\code{
$a=rand(5,9,2);
$inc=int(rint(3)+2);
$b=$a-$inc;
$ans=int(10^$inc);
}
% The above code chooses a random value $a between 5 and 9, to two
% significant figures. $b will be 2, 3, or 4 less than $a. \end{question}
```

Constants of Integration Questions in LaTeX

Question Mode: Formula mod C

The **Formula Mod C** question type is similar to the **Formula** type. Answers can be any kind of formula, entered in calculator syntax, but responses are graded as correct if they differ from the given answer by a constant value. Answers can include variables and standard functions. This question type is designed for indefinite integral questions.

Note:

- The **\answer** field must be in calculator syntax, not in TeX. Enter the answer as you expect the student to enter it (although equivalent answers are graded correct). Do not include + *C* in the answer.

Example 1

```
\begin{question}{Formula mod C}
\qutext{Calculate $\int 6x^2\, dx$.}
% Use calculator syntax for the answer, not TeX.
\answer{2x^3}
% Note: The response '2x^3+1' is also graded correct.
\end{question}
```

Example 2

```
\begin{question}{Formula mod C}
% This question uses randomized variables.
\qutext{Calculate $\int \var{a}x+\var{b}\, dx$.}
```

```
\answer{\var{a2}x^2+\var{b}x}
\code{
$a1=int(rint(9)+2);
$t=rint(2);
$a2=int(switch($t,$a1,-$a1));
$a=int(2*$a2);
$b=int(rint(20)+1);
}
% The above code chooses random integer coefficients a and b.
% a is an even integer with absolute value between 4 and 20.
% b is a positive integer between 1 and 20.
\end{question}
```

Equation Questions in LaTeX

Question Mode: Equation

The **Equation** question type accepts an equation (two formulas separated by an = sign) as an answer.

Notes

- One side of the correct answer must be in the form of only a single variable (for example, **y=...** or **...=z**). However, the student response does not need to be in this form. Any equivalent equation is graded correct.
- The **\answer** field must be in calculator syntax, not in TeX.

Example 1

```
\begin{question}{Equation}
\qutext{Find the equation of the line that passes though the points
$(-3,4)$ and $(6,1)$.}
% One side of the answer given must be a single variable.
\answer{y=-x/3+3}
% Note: The response 'y-1=-(x-6)/3' is also graded correct.
\end{question}
```

Example 2

```
\begin{question}{Equation}
% This question uses randomized variables.
\qutext{Find the equation of the line that is parallel to the line
$y=\var{m}x+\var{b}$ and that passes through the point
$(\var{c},\var{d})$.}
\answer{y=\var{m}x\var{sgn}\var{e}}
\code{
$m1=int(rint(19)+2);
$t=rint(2);
$m=int(switch($t,$m1,-$m1));
$b=int(rint(20)+1);
$c=int(rint(21)-10);
$d=int(rint(21)-10);
$e1=int($d-$m*$c);
$sgn=switch(lt($e1,0),"+","-");
$e=int(abs($e1));
}
% The above code chooses random integer coefficients m and b, and a
% random point (c,d). The y-intercept e is calculated. Care is
% taken to display the answer correctly because e may be positive or
% negative.
\end{question}
```

Formula List Questions in LaTeX

Question Mode: Formula List

The **Formula List** question type accepts an ordered list of numbers or formulas separated by commas.

Notes

- This is an *ordered* list. The entries in the student response (numbers or formulas) must appear in the same order as in the correct answer.
- This type is graded in the same manner as the **Ntuple** type.

Example 1

```
\begin{question}{Formula List}
\qutext{Compute the exact values of $\sin(\pi/3)$, $\cos(\pi/3)$, and
$\tan(\pi/3)$.}
% Separate the entries of the answer with commas.
\answer{sqrt(3)/2,1/2,sqrt(3)}
\end{question}
```

Example 2

```
\begin{question}{FormulaList}
% This question uses randomized variables.
\qutext{Find the slope $m$ and the vertical intercept $b$ of the line with the equation
$y=\var{m}x+\var{b}$.}
\answer{\var{m},\var{b}}
\code{
$m1=int(rint(9)+2);
$t=rint(2);
$m=int(switch($t,$m1,-$m1));
$b=int(rint(20)+1);
}
% The above code chooses random integer coefficients m and b.
% m is an integer with an absolute value between 2 and 10.
% b is a positive integer between 1 and 20.
\end{question}
```

See Also:

Coordinates of Points and Vectors Questions in LaTeX (page 280)

Formula Questions in LaTeX

Question Mode: Formula

The **Formula** question type is the most general math question type. Answers can be any kind of formula, entered in calculator syntax, or using the system **Symbolic Editor** entry mode. Independent of entry method, answers are encoded as MathML, enabling the system to grade equivalent answers correctly. Answers can include variables and standard functions.

Notes

- The **\answer** field is a calculated field. It must be in calculator syntax, not in TeX. Enter the answer as you expect the student to enter it (although equivalent answers are graded correct).
- If the answer is a number (that is, does not have any variables), it is recommended that you specify in the question text whether an exact response or an approximation is required. In the case of an approximation, the margin of error must be specified in your correct answer. This can be done by:
  - Specifying a margin of error using the **?** operator

For example, if the answer field were

```
{2.34 ? 0.01}
```

any response between **2.33** and **2.35** (inclusive) would be graded correct. In other words, the number following the **?** operator is the margin of error. For more information on tolerances, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

Example 1

```
\begin{question}{Formula}
\qutext{What is the derivative of $x^3+3\cos(x)-1$?}
% Use calculator syntax for the answer, not TeX.
\answer{3x^2-3*sin(x)}
\end{question}
```

Example 2

```
\begin{question}{Formula}
\qutext{Compute the exact value of $f'(2)$ if $f(x)=x^3+3\cos(x)-1$.}
\answer{12-3*sin(2)}
% Any numerical approximation is graded incorrect.
\end{question}
```

Example 3

```
\begin{question}{Formula}
\qutext{Compute $f'(2)$ if $f(x)=x^3+3\cos(x)-1$. Round your answer to 5 significant digits.}
\answer{9.2721}
% In this case, a more (or less) accurate response is graded incorrect.
\end{question}
```

Example 4

```
\begin{question}{Formula}
\qutext{Compute $f'(2)$ if $f(x)=x^3+3\cos(x)-1$. Your answer should be
correct to an accuracy of at least 5 decimal places.}
\answer{9.27211 ? 0.00001}
% Any response between 9.27210 and 09.27212 is graded correct.
% The response '12-3*sin(2)' is also acceptable. \end{question}
```

Example 5

```
\begin{question}{Formula}
% This question uses randomized variables.
\qutext{Find the formula for the inverse of the function
$f(x)=\var{a}x+\var{b}$.}
\answer{(x-\var{b})/\var{a}}
\code{
$a=int(rint(9)+2);
$b=int(rint(9)+2);
}
% The above code chooses random integer coefficients a and b between
% 2 and 10. \end{question}
```

Maple-graded Questions in LaTeX

Question Mode: Maple

The **Maple** question type allows you to create questions that are graded using Maple code. Specify the grading code in the **\maple** field. For information about grading questions using Maple code, see *Maple-graded Question Type - Overview (page 140)*.

Note: Because student responses are graded by Maple, there is no **\answer** field.

The **Maple** question type has two subtypes:

- *Maple Formula (page 273)*
- *Maple Syntax (page 273)*

Maple Formula

Using the **Maple Formula** subtype (**type = formula**), you can create *formula* questions. A student can use Maple T.A. syntax, which is converted to Maple syntax before passing the response to Maple for grading. For more information, see Maple Formula.

Example 1

```
\begin{question}{Maple}
\qutext{What is the derivative of $x^2$?}
\type{formula} % to allow the student to enter "2x"
\allow2d{1}
\maple{diff(x^2,x)}
\end{question}
```

Maple Syntax

Using the **Maple Syntax** subtype, you can create *general* questions. All student responses must use Maple syntax. For more information, see *Maple Syntax (page 143)*.

Example 2

```
\begin{question}{Maple}
\qutext{Find the solution to the initial-value
problem $xy'=\sin x$, $y(0)=0$. Give your answer
as the formula for $y(x)$.}
\allow2d{0}
\maple{Si(x);}
% The response "int(sin(t)/t,t=0..x)" is graded correct
\end{question}
```

Maple Plotting

In any Maple-graded question, you can plot the student response, or a function related to the response, using Maple plot code. Specify the plotting code in the **\plot** field.

In any question, including non-mathematical questions, you can include a Maple plot. Specify the Maple plot code in the **\drawMaplePlot** field.

Example 3

```
\begin{question}{maple}
\qutext{The graph of a trigonometric function $f(x)$
is given below.
\drawMaplePlot{plot(2*sin(Pi*x + Pi/4), x = -2..2);}
The graph passes through the point $(1/4,2)$.\\
\\
Give a formula for $f(x)$.}
\maple{2*sin(Pi*x + Pi/4);}
\plot{plot($RESPONSE, x = -2..2);}
% You would not use \plot for this
% question because it gives away the answer
\end{question}
```

Matrix Questions in LaTeX

Question Mode: Matrix

The **Matrix** question type lays out an n by m matrix of text fields, each of which accepts a number or formula. **Matrix** questions must be composed in LaTeX.

Notes

- The given correct answer must be presented as a comma-separated list with the first row listed first, then the second row, etc.
- The required **\size** macro gives the size of the answer matrix. The format is **\size{m}{n}**, where **m** is the number of rows and **n** is the number of columns.
- Maple Syntax questions can also be used for questions that will have a matrix response, and provide more flexibility than the Matrix questions in question authoring and grading.

Example 1

```
\begin{question}{Matrix}
\qutext{Find the transpose of the matrix
$$A=\left[\begin{array}{ccc} 1 & 2 & 3\\ 0 & 1 & -2\end{array}\right]$$}
% Specify the number of rows of the answer, and then the number of columns.
\size{3}{2}
% Use a comma-separated list, starting with the first row:
\answer{1,  0,
        2,   1,
        3,  -2}
% The answer is formatted for readability, but this is not required.
\end{question}
```

Example 2

```
\begin{question}{Matrix}
% This question uses randomized variables.
\qutext{Find the transpose of the matrix
$$A=\left[\begin{array}{ccc} \var{a} & \var{b} & \var{c}\\
\var{d} & \var{e} & \var{f}\end{array}\right]$$}
\size{3}{2}
\answer{\var{a}, \var{d},
        \var{b},  \var{e},
        \var{c},  \var{f}}
\code{
$a=int(rint(19)-9);
$b=int(rint(19)-9);
$c=int(rint(19)-9);
$d=int(rint(19)-9);
$e=int(rint(19)-9);
$f=int(rint(19)-9);
}
% The above code chooses six random integer coefficients a, b, c, d, e,
% and f, between -9 and 9.
\end{question}
```

Multi Formula Questions in LaTeX

Question Mode: Multi Formula

The **Multi Formula** question type accepts an *unordered* list of numbers or formulas separated by *semicolons*.

Notes

- This is an *unordered* list. The entries in the student response do not need to appear in the same order as in the correct answer.
- This type is commonly used for solutions of equations.
- This type can also be used for unordered lists of ntuples (questions requiring coordinates of points or vectors).
- A question that has a single formula answer is graded correct if you use this type. For example, consider the question of finding all solutions to a given equation that has only one solution, but is of a type that may have more. It is recommended that you use the **Multi Formula** type because the **Formula** type indicates to the student that there is only one solution.

Example 1

```
\begin{question}{Multi Formula}
\qutext{Find all roots of the polynomial $x^2+2x-24$.}
% Separate multiple entries of the answer with semicolons.
\answer{-6;4}
% Note: The response '4;-6' is graded correct.
\end{question}
```

Example 2

```
\begin{question}{Multi Formula}
\qutext{Find all exact solutions of the following system of equations.
\begin{align*}
x^2+y^2-4x&=0\\
x+y&=4
\end{align*}
}
% Note: The 'amsmath' package is needed to use the 'align' environment.
\answer{(2,2);(4,0)}
\end{question}
```

Example 3

```
\begin{question}{Multi Formula}
% This question uses randomized variables.
\qutext{Find all roots of the polynomial $x^2+\var{b}x-\var{c}$.}
\answer{\var{p};-\var{q}}
\code{
$p=int(rint(5)+1);
$q=int($p+rint(4)+2);
$b=int($q-$p);
$c=int($p*$q);
}
% The above code chooses random integers p and q, and then calculates
% the coefficients b and c. Note: b and c are both positive and
% b is not equal to 1.
\end{question}
```

Numeric Questions in LaTeX

Question Mode: Numeric

The **Numeric** question type accepts either a number in decimal or scientific notation, or a number with units for an answer.

You must specify units as an optional argument to the **\answer** macro. For example:

\answer[m/s^2]{9.8}

Because it is natural to specify the units after the number, you can use the format:

\answer{9.8}[m/s^2]

Notes:

- In the correct answer, the number must be enclosed in braces "**{ }**". The units must be enclosed in brackets "**[ ]**".
- If you do not specify units in the correct answer (that is, if you do not specify an argument to **\answer**), the system presents the student with a single text field in which to enter a response (without units). If you specify an argument to **\answer**, that argument is taken as the correct answer's units. In this case, the system displays two text fields. A student must enter the numerical part of the response in the first field and the units in the second field
- You must use recognized units.
- To define a custom list of units in a question bank, use the **\TableOfUnits** command in the LaTeX document's preamble (that is, before **\begin{document}** and after **\usepackage{ed}**). The **\TableOfUnits** command takes two required arguments, in the following order:
  tablename - table name, consisting of lower-case letters (**a**-**z**) with no spaces. Use the table name to reference the table in the document.
  units_list - the list of acceptable units. Units are defined with statements defining the units in terms of SI units or previously defined units. Each statement must end with a semicolon. For example:
  cm = 0.01m;
  mm = 0.1cm;
  The base SI units are:

  m (meter)

  kg (kilogram)

  s (second)

  A (amp)

  K (kelvin)

  cd (Candela)
- To use a custom list of units in a question, use the **\units{tablename}** command in the body of the question, where **tablename** is the table name (the first argument to **\TableOfUnits**).

Important: Because the system converts between systems of units using inexact conversion factors, it is recommended that you accept responses in a margin of error in questions with units.

- The **Numeric** question type has five distinct types of grading. Descriptions are given below, followed by usage instructions and examples.

1. **Exact value** - the student response must match the value given in the **\answer** field. There is no error tolerance. If you do not specify any other commands, this behavior is used.
2. **Exact value and number of significant digits** - the student response must agree with the value given in the **\answer** field *and* contain the specified number of significant digits. Specify the number of significant digits using the **\digits{n}** command, where **n** is the number of significant digits. For example, using **\digits{3}** with **\answer{2.5}**, only **2.50** is graded correct.
3. **Absolute tolerance** - the student response must agree with the value given in the **\answer** field to within a specified tolerance. Specify the absolute tolerance using the **\err{error}** command, where **error** is the absolute tolerance. For example, using **\err{0.25}** with **\answer{2.5}**, any response between **2.25** and **2.75** (inclusive) is graded correct. For example, **2.333333** is correct.
4. **Significant digits with tolerance** - the student response must agree with the value given in the **\answer** field to within a specified tolerance in the **nth** significant digit. Specify the tolerance using the **\err{k}** and **\digits{n}** commands, where **k** is the tolerance in the **nth** significant digit. For example, using **\err{2}** and **\digits{3}** (tolerance of **2** at the third significant digit) with **\answer{2.5}**, any response between **2.48** and **2.52** (inclusive) is graded correct. For example, **2.4999999** is correct.
5. **Relative tolerance** - the student response must agree with the value given in the **\answer** field to within a specified percent error. Specify the percent tolerance using the **\perc{error}** command, where **error** is the percent. For example, using **\perc{5}** with **\answer{2.5}**, any response between **2.5 * 0.95** and **2.5 * 1.05** (inclusive) is graded correct. For example, **2.388888** is correct.

- Consider the following recommendations when using this question type.

1. In general, allow some tolerance in a student response.
2. Provide a correct answer that is as accurate as possible.
3. If the use of units is not an important part of the question (that is, you are not testing the student's ability to convert units), do not include units in the **\answer** field. In this case, instruct the students to enter a purely numeric response in terms of a unit you specify.

Example 1

```
\begin{question}{numeric}
\qutext{Compute the exact value of the function $f(x)=3x^4-2x^2-1$  at
$x=2.1$.}
\answer{48.5243} % exact value grading
\end{question}
```

Example 2

```
\begin{question}{numeric}
\qutext{Compute the value of $\log(45)$. Round your answer to 5
significant digits.}
\answer{1.6532} % This question uses exact grading because  \digits is not used
% In this case, a more accurate response is graded incorrect.
\end{question}
```

Example 3

```
\begin{question}{numeric}
\qutext{Compute the value of $\log(45)$. Your answer must be correct
to at least 5 decimal places.}
\answer{1.653212514}
\err{0.00001} % absolute tolerance
% Any numerical response between 1.653202514 and 1.653222514 is
% graded correct.
\end{question}
```

Example 4

```
\begin{question}{numeric}
% This question uses algorithmic variables.
\qutext{Compute the value of $\log(\var{a})$. Your answer must be
correct to at least 5 decimal places.}
\answer{\var{ans}}
\err{0.00001}
\code{
$a=rand(1,100,4);
$ans=log($a);
}
% The above code chooses a random value a between 1 and  100, to four
% significant figures. The value of log(a) is then calculated.
\end{question}
```

Example 5

```
\begin{question}{numeric}
\qutext{A man is on the top of a tower that is 800 feet above ground.
How far can he see?
\newline\newline
Use 3960 miles for the radius of the earth.\newline
Express your answer to an accuracy of at least 6 significant digits.}
\answer{34.64134750325308}[mi]
\err{0.0001} % Note: Using \digits{6} would allow no tolerance, and would
             %  cause '182906 ft' and '34.6413 mi' to be graded incorrect
             %  because the values do not match the correct answer.
\end{question}
```

Example 6

```
\begin{question}{numeric}
% If the ability to convert units is not an important part of the
% question, do not use units.
\qutext{A cat is on the top of a tower that is 800 feet above ground.
How far (in miles) can it see?
\newline\newline
Use 3960 miles for the radius of the earth.\newline
Express your answer to an accuracy of at least 4 decimal places.\newline
Do NOT include any units in your answer.}
\answer{34.64134750325308}
\err{0.0001}
\end{question}
```

Example 7

```
\begin{question}{numeric}
% An algorithmic version of the previous question.
% The height is a multiple of 50 in [400,900).
\code{$height = int(range(400,900,50));
      $answer  = sqrt((3960 + $height/5280)^2 - 3960^2);}
\qutext{A woman is on the top of a tower that is \var{height}  feet above
ground. How far (in miles) can she see?
\newline\newline
Use 3960 miles for the radius of the earth.\newline
Express your answer to an accuracy of at least 6 significant digits.}
\answer[mi]{\var{answer}}
\err{2}    %  Accept any response within +/- 2 units
\digits{6} % in the sixth significant digit.
\end{question}
```

Restricted Formula Questions in LaTeX

Question Mode: Restricted Formula

The **Restricted Formula** question type is similar to the **Formula** type, but answers can use only the standard arithmetic operations (+, -, *, /, and ^) and the **sqrt()** function. The correct answer and student responses cannot contain logarithmic and trigonometric functions such as **log()** and **sin()**. Answers can include variables.

Notes

- The **\answer** field must be in calculator syntax, not in TeX. Enter the answer as you expect the student to enter it (although equivalent answers are graded correct).
- If the answer is a number (that is, does not have any variables), it is recommended that you specify in the question text whether an exact response or an approximation is required. In the case of an approximation, the margin of error must be specified in your correct answer. This can be done by:
  - Specifying a margin of error using the **?** operator

  For example, if the answer field were

  ```
  {2.34  ? 0.01}
  ```

  any response between **2.33** and **2.35** (inclusive) would be graded correct. In other words, the number following the **?** operator is the margin of error. For more information on tolerances, see *Setting a Margin of Error in Non-numeric Questions (page 184)*.

Example 1

```
\begin{question}{Restricted Formula}
\qutext{What is the exact value of $\sin(\frac{\pi}{4})$?}
% Use calculator syntax for the answer, not TeX.
\answer{1/sqrt(2)}
% Note: The response 'sin(pi/4)' would be graded correct in
% 'Formula' mode, but not in 'Restricted Formula' mode. \end{question}
```

Example 2

```
\begin{question}{Restricted Formula}
\qutext{Compute the value of $\sin(\frac{\pi}{4})$. Round your answer to 5 significant digits.}
\answer{0.70711}
% In this case, a more accurate response is graded as
% incorrect.
\end{question}
```

Example 3

```
\begin{question}{Restricted Formula}
\qutext{Compute the value of $\sin(\frac{\pi}{4})$. Your answer must
be correct to at least 5 decimal places.}
\answer{0.707107 ? 0.00001}
% Any numerical response between 0.707097 and 0.707117 is graded
% correct.
\end{question}
```

Example 4

```
\begin{question}{Restricted Formula}
% This question uses randomized variables.
\qutext{What is the exact value of $\sin(\frac{\var{n}\pi}{4})$?}
\answer{\var{s}1/sqrt(2)}
\code{
$m=rint(7);
$n=int(2*$m+3);
```

```
$s=switch($m,"","-","-","","","-","-");
}
% The above code chooses the random integer coefficient n to be an odd
% integer between 3 and 15. The 'switch' statement selects the correct
% sign for the answer. \end{question}
```

Coordinates of Points and Vectors Questions in LaTeX

Question Mode: Ntuple

The **Ntuple** question type accepts an ordered of numbers or formulas separated by commas. It is designed for questions that accept vector or point coordinates as an answer.

Notes

- This is an *ordered* list. The entries in the student response must appear in the same order as in the correct answer.
- The correct answer given should have parentheses, for example, **(3,5)** or **(2,1,-3)**. This is indicated in the instructions given under the answer box. However, a response *without* the parentheses is graded correct.
- This question type is graded in the same manner as the **Formula List** type. The only difference is the instructions displayed.

Example 1

```
\begin{question}{Ntuple}
\qutext{Find the intersection of the lines $y=x+1$ and $y=2-x$.}
% Separate the entries of the answer with commas. Parentheses
% should be included.
\answer{(1/2,3/2)}
% Note: The response "1/2,3/2" is graded correct.
\end{question}
```

Example 2

```
\begin{question}{Ntuple}
% This question uses randomized variables.
\qutext{Find the exact solution of the following system of equations.
\begin{align*}
\var{a}x+\var{b}y&=\var{e}\\
\var{c}x+\var{d}y&=\var{f}
\end{align*}
\newline Your answer should be an ordered pair $(x,y)$.}
\answer{(\var{ans1},\var{ans2})}
% Note: The 'amsmath' package is needed to use the 'align' environment.
\code{
$aa=int(rint(5)+2);
$b=int(rint(5)+2);
$c=int(rint(5)+2);
$d=int(rint(5)+2);
$e=int(rint(5)+2);
$f=int(rint(5)+2);
$det1=int($aa*$d-$b*$c);
$a=int(if($det1,$aa,$aa+1));
$det=int($a*$d-$b*$c);
$x=int($e*$d-$f*$b);
$y=int($a*$f-$c*$e);
$ansx=$x/$det;
$ansy=$y/$det;
$ansxint=int($x/$det);
$ansyint=int($y/$det);
$detabs=int(abs($det));
$xabs=int(abs($x));
$yabs=int(abs($y));
```

```
$numx=int(if(int(lt($ansx,0)),-$xabs,$xabs));
$numy=int(if(int(lt($ansy,0)),-$yabs,$yabs));
$ans1=if(int(eq($ansx,$ansxint)),"$ansxint","$numx/$detabs");
$ans2=if(int(eq($ansy,$ansyint)),"$ansyint","$numy/$detabs");
}
% The above code chooses random integer coefficients a, b, c, d, e,
% and f. a is modified if necessary so that the determinant of the
% system is nonzero. The solution is calculated using
% Cramer's Rule. Care is taken to display the answer
% coordinates correctly.
\end{question}
```

See Also:

Formula List Questions in LaTeX (page 271)

12 Plain Text Script Authoring

12.1 Using Script Files to Author Questions

You can author and edit all question types using the system plain text script file syntax. (To create advanced questions (*Advanced Question Types (page 178)*) and other problems that use the system API to produce interactive content, you cannot use the Question Editor.)

To create or edit script files for a question bank, use:

- A text or HTML editor
- LaTeX. See *LaTeX Authoring Overview (page 247)*

For some question types, you must use LaTeX.

Note: You can edit the script file for a question (not question bank) in the system **Question Editor** using the **Edit Source** functionality. It is recommended that you use the **Question Editor Edit Source** functionality only for minor editing.

To edit a question bank that was created in the **Question Editor** in a text or HTML editor, export the question bank to your hard drive. You can export any question bank in its plain text script format as a file with a **.qu** file extension.

After editing the plain text script file, import it and save the question bank into your class web site.

See Also:

Topic Structure within Question Banks (page 118)

Exporting Work to Your Hard Drive (page 116)

Opening a Saved Question Bank File (page 120)

12.2 Library of Example Scripts

This section provides example script files for:

Algorithmically generated figure labels inserted in your static reference figure—*Creating Algorithmic Graphic Labels (page 284)*

Algorithmically generated math fomulas—*Algorithmic Mathematical Formula Question Example Script (page 284)*

Algorithmically generated multiple selection questions—*Algorithmic Multiple Selection Question Example Script (page 284)*

Questions with embedded applets—*Applet Questions (page 288)*

Questions with plots—*Authoring Questions with Embedded Plots Example Script (page 297)*

Questions with tolerance in answers—*Answer Tolerance Example Script (page 285)*

Clickable Image questions—*Clickable Image Question Example Script (page 288)*

Essay questions—*Essay Question Example Script (page 289)*

Fill-in-the-blank questions—*Fill-in-the-Blank Question Example Script (page 289)*

Key word or phrase questions—*Key Word or Phrase Question Example Script (page 291)*

List questions—*List Question Example Script (page 292)*

Maple Formula questions—*Maple-graded Formula Question Example Script (page 293)*

Maple Syntax questions—*Maple Syntax Question Example Script (page 294)*

Multipart questions—*Multipart Question Example Script (page 295)*

Multiple choice questions—*Multiple Choice Question Example Script (page 296)*

Multiple selection questions—*Multiple Selection Question Example Script (page 296)*

Numeric questions—*Numeric Mathematical Questions Example Script (page 286)*

Question Designer questions—*Question Designer Question Example using Various Embedded Questions (page 298)*

Short phrase questions—*Short Phrase Question Example Script (page 300)*

True-or-False questions—*True-or-False Question Example Script (page 300)*

To use an example script:

1. Save the question definition as a text file with the extension **.qu**.
2. Import the text file.

Creating Algorithmic Graphic Labels

To author questions with algorithmically generated labels in figures, you must use LaTeX.

See Also:

Labeling Graphics in LaTeX Questions (page 263)

Algorithmic Mathematical Formula Question Example Script

The following example script creates a mathematical **Formula** question that has algorithmically generated variable data .

```
qu.1.topic=Formula Question Example Script@
qu.1.1.mode=Formula@
qu.1.1.name=<b>Traditional formula</b>@
qu.1.1.editing=useHTML@
qu.1.1.algorithm=
$n=int(rint(20) + 2);
@
qu.1.1.question=
  Let <math><mi>y</mi><mo>=</mo><msup><mi>x</mi><mn>$n</mn></msup></math>. What is the derivative
 of <math><mi>y</mi></math>  with respect to <math><mi>x</mi></math>?
@
qu.1.1.answer=$n x^($n - 1)
@
```

See Also:

Overview of Algorithmic Question Capabilities (page 217)

Algorithmic Multiple Selection Question Example Script

The following example script creates a **Multiple Selection** question with algorithmically generated data.

```
qu.1.topic=Multiple Selection Question Example Script@
qu.1.1.mode=Multiple Selection@
qu.1.1.name=<b>Multiple selection</b>
@
qu.1.1.editing=useHTML@
qu.1.1.algorithm=
```

```
 $a=int(2 * rint(10) + 2);
 $b=int(2 * rint(10) + 2);
 $c=int(2 * rint(10) + 2);
 $d=int(2 * rint(10) + 2);
 $e=int(2 * rint(10) + 1);
 $f=int(2 * rint(10) + 1);
 condition: ne($a,$b,$c,$d)  ne($e,$f);
@
qu.1.1.question=
Which of these numbers are even?@
qu.1.1.answer=1, 2, 3, 4@
qu.1.1.choice.1=$a@
qu.1.1.choice.2=$b@
qu.1.1.choice.3=$c@
qu.1.1.choice.4=$d@
qu.1.1.choice.5=$e@
qu.1.1.choice.6=$f@
```

See Also:

Overview of Algorithmic Question Capabilities (page 217)

Answer Tolerance Example Script

The following questions define mathematical questions that accept student responses to a specified precision or margin of error. For more information on precision and margins of error, see *Controlling Answer Tolerance (page 183)*.

Formula Mathematical Questions

For formula question types (including subtypes), you must use the "**?**" operator to define a margin of error. The "**?**" operator functions as a "+/-" operator.

Example 1.1

You can specify an *absolute* margin of error using the form **x ? y**, which accepts **x +/- y**. This question grades as correct any student response between **0** and **4** (inclusive).

```
qu.1.topic=Non-numeric Math Questions@

qu.1.1.mode=Formula@
qu.1.1.name=Absolute margin of error@
qu.1.1.editing=useHTML@
qu.1.1.question=Enter a number within 2 units of  2.@
qu.1.1.answer=2?2@
qu.1.1.tolerance=1.0E-9@
```

Note:

- For formula questions, there is a **tolerance** of **1.0E-9**, which represents the floating-point limit in the system.

Example 1.2

You can specify a *percentage* margin of error using the form **x (1 ? y)**, where **100y** is the percentage tolerance, which accepts **x +/- x*y**. This question grades correct any response within **5%** of the correct answer.

```
qu.1.2.question=What is the length of the shadow cast by a 10 foot wall when the sun is 30
degrees above the horizon?@
qu.1.2.answer=10 / tan(pi/6) foot (1 ? 0.05)@
qu.1.2.tolerance=1.0E-9@
qu.1.2.mode=Dimensioned Formula@
qu.1.2.name=Percentage tolerance with units@
```

```
qu.1.2.comment=Using the formula 10 / tan(pi/6)  foot, the correct answer is 17.3 feet.@
qu.1.2.editing=useHTML@
```

Note:

- By using the **comment** field, you return custom feedback to a student if the response is incorrect.

Example 1.3

You can use a margin of error in questions that use algorithmic variables.

```
qu.1.3.mode=Formula@
qu.1.3.name=Algorithmic Variables@
qu.1.3.editing=useHTML@
qu.1.3.algorithm=$InitialSize=int(rint(2,10)*1000);
$DoublingTime=rint(5,10);
$NumDays=rint(2,4);
$DeltaTime=int($NumDays*24);
$Exponent=ln(2)/$DoublingTime;
$Size=int($InitialSize*e^($Exponent*$DeltaTime));
@
qu.1.3.question=A bacterial culture is initially of size $InitialSize. If the doubling time is
 $DoublingTime hours,
what is the size of the culture after $NumDays days?@
qu.1.3.answer=$Size ? 1@
qu.1.3.tolerance=1.0E-9@
```

For general information on algorithmic variables, see *Overview of Algorithmic Question Capabilities (page 217)*.

Example 1.4

You can use margins of errors with complicated answers. Enclose the answer in parentheses when there is a percentage margin of error. Otherwise, the percentage error multiplies only the last term in your answer.

```
qu.1.4.mode=Formula@
qu.1.4.name=Absolute margin of error in sum@
qu.1.4.comment=The total area is $AreaSquare +  $AreaCircle.@
qu.1.4.editing=useHTML@
qu.1.4.algorithm=$length=rand(2,15,2);
$diameter=rand(3,10,2);
$AreaSquare=$length*$length;
$AreaCircle=decimal(2,pi*($diameter/2)^2);
@
qu.1.4.question=What is the total area of a square with sides of length $length and a circle
with diameter $diameter?@
qu.1.4.answer=($length^2+pi*($diameter/2)^2) (1  ? 0.01)@
qu.1.4.tolerance=1.0E-9@
```

Important:

- The exact answer is enclosed in parentheses so that the margin of error applies to both terms in the sum. If the answer were **$length^2+pi*($diameter/2)^2 (1 ? 0.01)**, the margin of error would be based on only the second term, **pi*($diameter/2)^2**, because the assumed multiplication is of higher precedence than the addition.

Numeric Mathematical Questions

For **Numeric** question types, you cannot use the "**?**" operator to define a margin of error. It is recommended that you use the **Question Editor** on-screen dialog boxes for setting the margin of error in numeric and numeric-with-unit questions. However, once familiar with the syntax, you may prefer to use a text or HTML editor.

For more information on setting the precision and margin of error in the Question Editor, see *Setting Answer Tolerance in Numeric Questions (page 86)*.

Example 2.1

To specify that the student response must *exactly* match the *value* of your answer, use **grading=exact_value**.

```
qu.2.topic=Numeric Math Questions@

qu.2.1.question=Round 2.76789m to three significant digits.@
qu.2.1.answer.num=2.77@
qu.2.1.answer.units=m@
qu.2.1.showUnits=true@
qu.2.1.grading=exact_value@
qu.2.1.mode=Numeric@
qu.2.1.name=2.77m (exact)@
qu.2.1.editing=useHTML@
```

Notes:

- This question grades correct the response **2.77m** and equal-valued responses with trailing zeros, for example, **2.770m** and **2.7700m**.
- To specify the answer, assign the numeric component to **answer.num** and the units to **answer.units**.
- If you set **showUnits=false**, the system does not display the **Units** text field in the question.

Example 2.2

To specify that the student response must *exactly* match the *value* of your answer and contain specified number of significant digits, use **grading=exact_sigd** and **digit=<n>**, where <**n**> is a positive integer that defines the required number of significant digits.

```
qu.2.2.question=A person counts 162 cars driving up a street in 1.00 hour.
How many cars drive up the street per minute? Use the correct number of significant digits in
your answer.@
qu.2.2.answer.num=2.70@
qu.2.2.answer.units=@
qu.2.2.showUnits=true@
qu.2.2.grading=exact_sigd@
qu.2.2.digit=3@
qu.2.2.mode=Numeric@
qu.2.2.name=2.70 (exactly 3 significant digits)@
qu.2.2.editing=useHTML@
```

Notes:

- This question grades correct the response **2.70**, but grades incorrect equal-values responses with a different number of trailing zeros, for example, **2.7** and **2.700**.
- Ensure that <**n**> is greater than or equal to the number of digits in the correct answer (the value in the **answer.num** field). Otherwise, all student responses are graded incorrect.
- If the correct answer is dimensionless and you set **showUnits=true**, the system displays the question with a **Units** text field.

Example 2.3

To specify an *absolute* margin of error, use **grading=toler_abs** and **err=<tolerance>**, where **<tolerance>** is a floating-point number. This question grades correct any answer within **0.001** of **3.14159**.

```
qu.2.3.question=Estimate pi to at least 3 decimal places.@
qu.2.3.answer.num=3.14159@
qu.2.3.answer.units=@
qu.2.3.showUnits=false@
qu.2.3.grading=toler_abs@
qu.2.3.err=0.001@
qu.2.3.mode=Numeric@
```

```
qu.2.3.name=Pi (+/- 0.001)@
qu.2.3.editing=useHTML@
```

Example 2.4

To specify a precision **and** an *absolute* margin of error as a value in a significant digit, use **grading=toler_sigd**, **digit=<n>**, and **err=<tolerance>**, where **<n>** is a positive integer specifying the number of significant digits required and **<tolerance>** is a floating-point number specifying the tolerance value in the **<n>**th digit. This question grades correct **3.141** and **3.142** because they contain **4** significant digits **and** are within **0.001** of **3.14159**. All other responses are graded incorrect.

```
qu.2.4.question=Estimate pi to four significant digits (to within a unit in the fourth significant
 digit).@
qu.2.4.answer.num=3.14159@
qu.2.4.answer.units=@
qu.2.4.showUnits=false@
qu.2.4.grading=toler_sigd@
qu.2.4.digit=4@
qu.2.4.err=1.0@
qu.2.4.mode=Numeric@
qu.2.4.name=Pi (+/- 1 in the 4th significant digit)  using four significant digits@
qu.2.4.editing=useHTML@
```

Notes:

- The behavior of this question is different from that of **Example 2.3**, which has only a margin of error, but no required precision.
- The margin of error is applied with respect to the value in the **answer.num** field, not the result of rounding the value in the **answer.num** field to **<n>** significant digits. For example, **3.14159** rounded to **4** significant digits is **3.142**. However, **3.143** is graded incorrect because it is not within **0.001** of **3.14159**.

Example 2.5

To specify a *percentage* margin of error, use **grading=toler_perc** and **perc=<percentage>**, where **<percentage>** is the percentage margin of error. This question grades correct any answer within **0.1%** of **3.14159**.

```
qu.2.5.question=Estimate pi to within 1%.@
qu.2.5.answer.num=3.14159@
qu.2.5.answer.units=@
qu.2.5.showUnits=false@
qu.2.5.grading=toler_perc@
qu.2.5.perc=0.1@
qu.2.5.mode=Numeric@
qu.2.5.name=Pi (+/- 0.1%)@
qu.2.5.editing=useHTML@
```

Applet Questions

Your question banks can integrate interactive learning objects in the form of Java™ applets. For more information, see *Advanced Question Types (page 178)*.

Clickable Image Question Example Script

The following example script creates a **Clickable Image** question.

```
qu.1.topic=Clickable Image Question Example Script@
qu.1.1.mode=Clickable Image@
qu.1.1.name=Locate Trachea@
qu.1.1.editing=useHTML@
qu.1.1.question=Click on the <b><i>trachea</i></b>  in the diagram below.
<p>@
```

```
qu.1.1.imageURL=/questionbanks/physio/respsys.gif@
qu.1.1.answer=1@
qu.1.1.width=578@
qu.1.1.height=394@
qu.1.1.region.1=288,198,278,235,278,270,289,272,290,237,300,203@
qu.1.1.region.2=293,166,302,171,311,178,309,186,306,195,299,202,288,194,
292,188,293,174@
qu.1.1.region.3=305,227,300,238,294,291,296,309,312,327,328,345,332,368,
355,378,371,383,378,355,376,311,367,282,353,256,317,229@
qu.1.1.region.4=256,225,268,232,271,244,267,260,274,277,280,303,274,320,
275,344,268,358,239,376,192,377,189,358,190,331,197,283,205,270,224,244@
```

See Also:

Clickable Image Questions (page 154)

Essay Question Example Script

The following example script creates an **Essay** question.

```
qu.1.topic=Essay Question Example Script@
qu.1.1.mode=Essay@
qu.1.1.name=Friction@
qu.1.1.editing=useHTML@
qu.1.1.question=Contrast <i>kinetic</i>  and <i>static</i> friction.@
```

See Also:

Ungraded Essay Questions (page 155)

Fill-in-the-Blank Question Example Script

The following example script creates a fill-in-the-blank (**Blanks**) question. This question uses the *in-line* style of presenting the student response entry cell. To place the student response entry cell on a separate line following the question, use the Short Phrase question mode.

```
qu.1.topic=Blanks Question Example Script@
qu.1.1.mode=Blanks@
qu.1.1.question=
  Select  "red" from the drop-down box. <1>
  <br>
  Enter  "red" or "green" in the text field. <2>
  <br>
  Enter  3 or 4 in the formula box. <3>
@
qu.1.1.blank.1=red,green,blue@
qu.1.1.blank.2=red,green@
qu.1.1.blank.3=3,4@
qu.1.1.grader.1=menu@
qu.1.1.grader.2=text@
qu.1.1.grader.3=formula@
```

Notes

- To display multiple entry cells, use the **<n>** structure in the text of the question (where **n** corresponds to the number of the entry cell).
- The drop-down box in **Blank 1** allows the student to select from a list of options (**red**, **green**, or **blue**).
- The text field in **Blank 2** requires that the student enter a text response that exactly matches the correct answer. Note: The **text** question subtype can contain a list of correct answers to allow for more than one correct answer.
- The formula text field in **Blank 3** requires that the student enter a formula or number without units.

See Also:

Blanks Questions (page 153)

Formula (page 137)

Rules for Grading Free Response Questions (page 158)

Graph Plotting Example Script

The following example script creates an **Multi Formula** question with a graph.

```
qu.1.1.mode=MultiFormula@
qu.1.1.question=<p class="noindent">What are the <math
xmlns="http://www.w3.org/1998/Math/MathML"  display="inline">
  <mi>x</mi>
 </math>-intercepts
of the graph shown?
</p>
<applet code="applets.grapher.Graph"  width="250" height="250" archive="graphing.jar">
 <param name="y1" value="(x-1)(x+3)"/>
 <param name="gridLines" value="12"/>
 <param name="xMin" value="-6"/>
 <param name="xMax" value="6"/>
 <param name="yMin" value="-6"/>
 <param name="yMax" value="6"/>
</applet>@
qu.1.1.answer=1;-3@
qu.1.1.info=@
```

See Also:

Graph Plotting in LaTeX Questions (page 259)

Graph Sketching Question Example Script

The following example scripts create **Sketch** questions.

```
qu.1.1.mode=sketch@
qu.1.1.question=<p class="noindent">Sketch the graph of the function <math
xmlns="http://www.w3.org/1998/Math/MathML"  display="inline">
  <mi>y</mi>
  <mo>=</mo>
  <mi>x</mi>
  <mo>+</mo>
  <mn>1</mn>
 </math>.</p>@
qu.1.1.gridlines=4@
qu.1.1.axes=-2,2,-2,2@
qu.1.1.axes.labeled=true@
qu.1.1.axes.background=\@ImageBase @
qu.1.1.example=-2,-1 1,2@
qu.1.1.answer=check(( linear($1) ) &&  ( goes_through($1,0,1) ))@
qu.1.1.info=@

qu.1.2.mode=sketch@
qu.1.2.algorithm=$a = int(rand(2,6));
$a2 = int($a*$a);
$xmax = int(2*$a);
$x = sqrt(2)*$a;@
qu.1.2.question=<p class="noindent">Sketch the graph of the function <math
xmlns="http://www.w3.org/1998/Math/MathML"  display="inline">
  <mi>y</mi>
  <mo>=</mo>
```

```
  <msup>
   <mrow>
    <mi>x</mi>
   </mrow>
   <mrow>
    <mn>2</mn>
   </mrow>
  </msup>
  <mo>&#8725;</mo>
  <mn>${a2}</mn>
 </math>.</p>@
qu.1.2.gridlines=4@
qu.1.2.axes=-${xmax},${xmax},-4,4@
qu.1.2.axes.labeled=true@
qu.1.2.axes.background=\@ImageBase @
qu.1.2.example=-${x},2 -${a},1 0,0 ${a},1  ${x},2@
qu.1.2.answer=check(( goes_through($1,0,0)  && goes_through($1,-${a},1) &&
goes_through($1,${a},1)  ) && ( slope_at($1,0) == 0 ) && (
decreasing(restriction($1,-${x},0))  ) && ( increasing(restriction($1,0,${x})) ) && (
concave_up($1)  ))@
qu.1.2.info=@
```

See Also:

Graph Sketching Questions in LaTeX (page 260)

Key Word or Phrase Question Example Script

There is no **Question Editor** interface for **Key Words** (or phrase) questions. To create a key word question, you must use a text or HTML editor, or LaTeX to produce script files

In **Key Words** questions, the text of the student response is checked for key words. The system ignores other text, extra white space, and capitalization in the student response.

In the correct answer, identify keywords by enclosing them in parentheses. The complete text of your correct answer (omitting parentheses) is used as feedback if the student response is incorrect.

The following example script creates a **Key Words** question.

```
qu.1.topic=Key Words Question Example Script@
qu.1.1.mode=Key Words@
qu.1.1.name=<b>Translation into French - four houses</b>@
qu.1.1.question=<b>Translate into French:</b>
<p>"four houses".</p>
@
qu.1.1.answer=
  The French phrase for "four houses"  is "(quatre)  (maisons)"
@
```

See Also:

Key Word or Phrase Questions (page 160)

Labeling Graphics in Questions Example Script

The following example scripts create an **Numeric** question with a labeled graph.

```
qu.1.5.mode=Numeric@
qu.1.5.question=<p class="noindent">A ball is thrown upward from the top of a 40 foot high
 tower at an initial velocity of 64
feet/second. The graph of the height of the ball versus time is shown below. What is the
maximum
height of the ball in feet? Express your answer to an accuracy of at least 3 significant
figures.
```

```
</p>
<applet code="applets.labelImage.LabelImage"  width="250" height="200">
 <param name="image" value="\@ImageBase ball.gif"/>
 <param name="label.1.text" value="time (seconds)"/>
 <param name="label.1.x" value="120"/>
 <param name="label.1.y" value="10"/>
 <param name="label.2.text" value="height (meters)"/>
 <param name="label.2.x" value="10"/>
 <param name="label.2.y" value="100"/>
 <param name="size" value="2"/>
</applet>@
qu.1.5.answer.num=104 ? 1@
qu.1.5.answer.units=@
qu.1.5.showUnits=false@
qu.1.5.info=@
qu.1.5.grading=exact_value@
```

List Question Example Script

Example 1: List Question - Textbox Display - Exact Grader Options

Sets a question with a text response cell that uses the exact grader to enforce capitalization and punctuation in the answer. Specifies three expected responses with credit for each.

```
qu.1.1.question=
  Who first developed the General Theory of Relativity?
@
qu.1.1.grader=exact@
qu.1.1.display=text@
qu.1.1.answer.1=Einstein@
qu.1.1.credit.1=1.0@
qu.1.1.answer.2=Albert Einstein@
qu.1.1.credit.2=1.0@
qu.1.1.answer.3=Stephen Hawking@
qu.1.1.credit.3=0.25@
qu.1.1.mode=List@
qu.1.1.editing=useHTML@
```

Example 2: List Question - Textbox Display - Relaxed Grader Options

Sets a question with a text response cell that uses the relaxed grader to ignore case sensitivity and punctuation in student responses. Specifies three expected responses with credit for each and a comment for each.

```
qu.1.2.question=
  Who first developed the General Theory of Relativity?
@
qu.1.2.grader=relaxed@
qu.1.2.display=text@
qu.1.2.answer.1=Einstein@
qu.1.2.credit.1=1.0@
qu.1.2.comment.1=Yes, but you should specify his first name too.@
qu.1.2.answer.2=Albert Einstein@
qu.1.2.comment.2=Exactly correct.@
qu.1.2.credit.2=1.0@
qu.1.2.answer.3=Stephen Hawking@
qu.1.2.credit.3=0.25@
qu.1.2.comment.3=Your answer is partially correct. Hawking helped to extend Albert Einstein's
General Theory of Relativity.@
qu.1.2.mode=List@
qu.1.2.editing=useHTML@
```

Example 3: List Question - Menu Display - Exact Grader Options

Sets a question with a menu list response object that uses the exact grader. Specifies credit for each option in the list, and a comment for each.

```
qu.1.3.question=
  Who first developed the General Theory of Relativity?
@
qu.1.3.grader=exact@
qu.1.3.display=menu@
qu.1.3.answer.1=Sir Isaac Newton@
qu.1.3.credit.1=0@
qu.1.3.comment.1=No, Sir Isaac Newton formed the theory of Newtonian gravitation that preceded
 Albert Einstein's General Theory of relativity.@
qu.1.3.answer.2=Albert Einstein@
qu.1.3.comment.2=Correct.@
qu.1.3.credit.2=1.0@
qu.1.3.answer.3=Stephen Hawking@
qu.1.3.credit.3=0.25@
qu.1.3.comment.3=Your answer is partially correct. Hawking helped to extend Albert Einstein's
General Theory of Relativity.@
qu.1.3.mode=List@
qu.1.3.editing=useHTML@
```

See Also:

List Questions (page 160)

Maple-graded Formula Question Example Script

The following example script creates a **Maple-graded** question using the **Formula** subtype.

```
qu.1.topic=Maple Formula Question Example Script@
qu.1.1.question=What is the derivative of x^2 with respect to x?@
qu.1.1.maple=evalb($RESPONSE = diff(x^2,x));@
qu.1.1.type=formula@
qu.1.1.mode=Maple@
qu.1.1.allow2d=1@
qu.1.1.comment=The correct answer is 2*x.@
qu.1.1.name=Derivative - Maple Formula@
qu.1.1.editing=useHTML@

qu.1.2.question=What is the integral of 2*x with respect to x?@
qu.1.2.maple=evalb(diff($RESPONSE,x)=2*x);@
qu.1.2.type=formula@
qu.1.2.mode=Maple@
qu.1.2.comment=The correct answer is x^2.@
qu.1.2.name=Integral - Maple Formula@
qu.1.2.editing=useHTML@
```

To return feedback to a student if the response is incorrect, you must use the **comment** field. For more information, see *Using the Comment Field in a Maple Question (page 294)*.

See Also:

Maple-graded Question Type - Overview (page 140)

Plotting a Student Response (page 187)

Tutorial: Maple-based Algorithmic Question Authoring (page 244)

Using the Comment Field in a Maple Question

The following example script returns feedback to the student including a comment in a Maple formula question.

Note: When a student response is incorrect, Maple question subtypes can return feedback. You must enter Maple code that evaluates to the correct answer by using the **$ANSWER** variable. See Maple-graded questions in the Question Editor.

```
qu.1.topic=Maple Comment Example Script@
qu.1.1.question=What is the greatest common divisor of  $a and $b?@
qu.1.1.maple=evalb($RESPONSE - $ANSWER)=0);@
qu.1.1.maple_answer=gcd($a, $b);@
qu.1.1.type=formula@
qu.1.1.mode=Maple@
qu.1.1.allow2d=1@
qu.1.1.comment=Try again!@
qu.1.1.editing=useHTML@
qu.1.1.algorithm=$a=int(rand(3, 100)); $b=int(rand(3,  100)); $gcd=maple(gcd($a, $b));@
```

See Also:

The Algorithm Designer (page 80)

Maple Syntax Question Example Script

The following example script creates a **Maple-graded** question using the **Syntax** subtype, and using the text-entry student response area.

```
qu.1.topic=Maple Syntax Question Example Script@
qu.1.1.question=What is the Maple syntax for the derivative of x^2 with respect to x? Do not
include the trailing semicolon.@
qu.1.1.maple=evalb("$RESPONSE" = "diff(x^2,x)") or evalb("$RESPONSE" = "Diff(x^2,x)");@
qu.1.1.type=maple@
qu.1.1.mode=Maple@
qu.1.1.allow2d=0@

qu.1.1.comment=The correct answer is diff(x^2,x).@
qu.1.1.name=Derivative - Maple Syntax@
qu.1.1.editing=useHTML@

qu.1.2.question=What is the Maple syntax for the integral of 2*x with respect to x? Do not
include the trailing semicolon.@
qu.1.2.maple=evalb("$RESPONSE"="int(2*x,x)") or evalb("$RESPONSE"="Int(2*x,x)");@
qu.1.2.type=maple@
qu.1.2.mode=Maple@
qu.1.1.allow2d=0@

qu.1.2.comment=The correct answer is int(2*x,x).@
qu.1.2.name=Integral - Maple Syntax@
qu.1.2.editing=useHTML@
```

To return feedback to a student if the response is incorrect, you must use the **comment** field. For more information, see *Using the Comment Field in a Maple Question (page 294)*.

See Also:

Maple-graded Questions in the Question Editor (page 96)

Maple-graded Question Type - Overview (page 140)

Plotting a Student Response (page 187)

Multipart Question Example Script

Multipart questions can consist of any combination of questions as parts, including nested multipart questions.

The following example script creates a **Multipart** question that contains a nested **Multipart** question.

```
qu.1.topic=Multipart Question Example Script@

qu.1.1.mode=Multipart@
qu.1.1.name=<b>Full multipart question featuring nested multipart</b>@
qu.1.1.editing=useHTML@
qu.1.1.algorithm=
$n=int(rand(5,15));
$a=int(rand(1,10));
$nminus = int($n  - 1);
$nplus = int($n  + 1);
$isEven=int(eq($n/2, int($n/2)));
$k=if($isEven,  1, 3);
$type=if($isEven,  "even", "odd");
$othertype=if($isEven,  "odd", "even");
@
qu.1.1.question=
  This question concerns the integral of the function
  <math>
  <msup>
  <mi>x</mi>
  <mn>$n</mn>
  </msup>
  </math>.
@
qu.1.1.weighting=1,1,1@
qu.1.1.numbering=alpha@
qu.1.1.part.1.question=
  In the first part, use a geometric argument to decide whether
  <math>
  <msubsup>
  <mo>&Integral;</mo>
  <mn>-$a</mn>
  <mn>$a</mn>
  </msubsup>
  <msup><mi>x</mi><mn>$n</mn></msup>  <mi>dx</mi>
  </math> is:
@
qu.1.1.part.1.editing=useHTML@
qu.1.1.part.1.answer=$k@
qu.1.1.part.1.mode=Multiple Choice@
qu.1.1.part.1.choice.3=zero@
qu.1.1.part.1.choice.2=negative@
qu.1.1.part.1.choice.1=positive@
qu.1.1.part.2.part.2.question=
  Use the indefinite integral above to compute
  <math>
  <msubsup>
  <mo>&Integral;</mo>
  <mn>-$a</mn>
  <mn>$a</mn>
  </msubsup>
  <msup><mi>x</mi><mn>$n</mn></msup>  <mi>dx</mi>
  </math>
@
qu.1.1.part.2.weighting=1,1@
qu.1.1.part.2.question=
  In the second part, check your answer by computing the integral.
@
qu.1.1.part.2.editing=useHTML@
```

```
qu.1.1.part.2.numbering=roman@
qu.1.1.part.2.part.1.mode=Formula Mod C@
qu.1.1.part.2.part.2.mode=Formula@
qu.1.1.part.2.part.1.editing=useHTML@
qu.1.1.part.2.part.2.answer=($a^$nplus - (-$a)^$nplus)  / $nplus@
qu.1.1.part.2.mode=Multipart@
qu.1.1.part.2.part.2.editing=useHTML@
qu.1.1.part.2.part.1.question=
  Evaluate
  <math>
  <mo>&Integral;</mo>  <msup><mi>x</mi><mn>$n</mn></msup>  <mi>dx</mi>
  </math>.
@
qu.1.1.part.2.part.1.answer=x^$nplus / $nplus@
qu.1.1.part.3.question=
  The geometric argument in the first part was based on the fact that
  <math>
  <msup>
  <mi>x</mi>
  <mn>$n</mn>
  </msup>
  </math> is an <1> function.
@
qu.1.1.part.3.extra=$othertype@
qu.1.1.part.3.editing=useHTML@
qu.1.1.part.3.blank.1=$type@
qu.1.1.part.3.mode=Complete Sentence@
```

See Also:

Multipart Questions (page 162)

Multiple Choice Question Example Script

The following example script creates a **Multiple Choice** question.

```
qu.1.topic=Multiple Choice Question Example Script@
qu.1.1.mode=Multiple Choice@
qu.1.1.name=Type of Point@
qu.1.1.editing=useHTML@
qu.1.1.question=For f(x)=(x-4)(x-1)(x+3), classify the point with x-coordinate 2/3+1/3*37^(1/2).@
qu.1.1.answer=2@
qu.1.1.choice.1=global minimum@
qu.1.1.choice.2=local minimum@
qu.1.1.choice.3=inflection point@
qu.1.1.choice.4=local maximum@
qu.1.1.choice.5=global maximum@
qu.1.1.choice.6=none of the above@
```

See Also:

Multiple Choice Questions (page 163)

Multiple Selection Question Example Script

The following example script creates a **Multiple Selection** question.

```
qu.1.topic=Multiple Selection Question Example Script@
qu.1.1.mode=Multiple Selection@
qu.1.1.name=Characterization of function at a point@
qu.1.1.editing=useHTML@
qu.1.1.question=Classify f(x)=x^3-3x^2+4 at the point with x-coordinate 2.
<p><i>(Choose all that apply)</i></p>@
qu.1.1.answer=2, 5@
```

```
qu.1.1.choice.1=negative@
qu.1.1.choice.2=zero@
qu.1.1.choice.3=positive@
qu.1.1.choice.4=increasing@
qu.1.1.choice.5=extremum@
qu.1.1.choice.6=decreasing@
qu.1.1.choice.7=concave down@
qu.1.1.choice.8=inflection point@
qu.1.1.choice.9=concave up@
```

See Also:

Multiple Selection Questions (page 164)

Authoring Questions with Embedded Plots Example Script

To author applet graphing questions, you must use a text or HTML editor, or LaTeX to produce script files. For details on using LaTeX to generate questions that display graphs, see *Graph Plotting in LaTeX Questions (page 259)*.

You can also embed Maple plots in questions using the **Question Editor**. For more information, see *Questions with Plots (page 148)*.

The following example script creates a question that generates graphs using algorithmically driven data.

```
qu.4.7.mode=Formula@
qu.4.7.name=Algorithmically generated graph@
qu.4.7.comment=
The correct answer is
<math>
 <msup><mi>x</mi><mn>2</mn></msup>
 <mo>-</mo>
 <mn>$B</mn>  <mi>x</mi>
 <mo>-</mo>
 <mn>$C</mn>
</math>
@
qu.4.7.editing=useHTML@
qu.4.7.algorithm=
 $a=int(rand(-8,5));
 $b=int(rand(-$a+2, 10));
 $yMin=5*int((-0.25($b-$a)^2)/5) - 5;
 $yMax=-$yMin;
 $B=int($a+$b);
 $C=int(-$a*$b);
@
qu.4.7.question=
Give a formula for the quadratic equation shown in the graph below:
<p>
<center>
  <applet code="applets.grapher.Graph"  archive="graphing.jar"  width="250"  height="250">
  <param name="xMin"  value="-10">
  <param name="xMax"  value="10">
  <param name="yMin"  value="$yMin">
  <param name="gridLines"  value="10">
  <param name="yMax"  value="$yMax">
  <param name="y1" value="(x-$a)(x-$b)">
  </applet>
</center>
@
qu.4.7.answer=(x-$a)(x-$b)@
```

Question Designer Question Example using Various Embedded Questions

This is an example of a question designer question containing response objects within the question statement. **Note**: The Question Designer question type supersedes the Inline question type.

Description

- Question Designer questions are multipart questions.
- You control question weighting.
- You control formatting of the entire question (including tables).
- Blanks can be inserted inside tables (to create fill-in-the-blanks tables of data).
- Multiple Choice and Multiple Selection Questions are set with response choices displayed *horizontally* rather than following the question. This enables you to create survey-style questions.
- Individual response cells are indicated by angle brackets, in the form <#>, where # corresponds to the sequentially numbered part.
- This question includes individual response objects (embedded questions) as follows.
  - Blank 1 - **Formula** question
  - Blank 2 - **Numeric** question
  - Blank 3 - **Numeric** question with required units
  - Blank 4 - **List** question, **text** input, **relaxed** grading, multiple answers, partial credit
  - Blank 5 - **List** question, **text** input, **exact** grading, multiple answers, partial credit
  - Blank 6 - **List** question, **menu** input, **exact** grading, partial credit

```
qu.1.topic=Question Designer Question Example@
qu.1.1.mode=Inline@
qu.1.1.editing=useHTML@
qu.1.1.weighting=1,1,1,1,1,1@
qu.1.1.numbering=alpha@
qu.1.1.part.1.editing=useHTML@
qu.1.1.part.1.tolerance=1.0E-9@
qu.1.1.part.1.question=xx@
qu.1.1.part.1.answer=$n x^($n2)@
qu.1.1.part.1.mode=Formula@
qu.1.1.part.2.editing=useHTML@
qu.1.1.part.2.question=xx@
qu.1.1.part.2.answer.num=11@
qu.1.1.part.2.mode=Numeric@
qu.1.1.part.2.showUnits=false@
qu.1.1.part.2.grading=exact_value@
qu.1.1.part.2.answer.units=@
qu.1.1.part.3.showUnits=true@
qu.1.1.part.3.editing=useHTML@
qu.1.1.part.3.question=xx@
qu.1.1.part.3.grading=exact_value@
qu.1.1.part.3.answer.units=ft/min@
qu.1.1.part.3.mode=Numeric@
qu.1.1.part.3.answer.num=10@
qu.1.1.part.4.grader=relaxed@
qu.1.1.part.4.editing=useHTML@
qu.1.1.part.4.question=xx@
qu.1.1.part.4.answer.3=Stephen Hawking@
qu.1.1.part.4.answer.2=Albert Einstein@
qu.1.1.part.4.answer.1=Einstein@
qu.1.1.part.4.mode=List@
qu.1.1.part.4.display=text@
```

```
qu.1.1.part.4.credit.3=0.5@
qu.1.1.part.4.credit.2=1.0@
qu.1.1.part.4.credit.1=1.0@
qu.1.1.part.5.grader=exact@
qu.1.1.part.5.editing=useHTML@
qu.1.1.part.5.question=xx@
qu.1.1.part.5.answer.3=Stephen Hawking@
qu.1.1.part.5.answer.2=Albert Einstein@
qu.1.1.part.5.answer.1=Einstein@
qu.1.1.part.5.mode=List@
qu.1.1.part.5.display=text@
qu.1.1.part.5.credit.3=0.5@
qu.1.1.part.5.credit.2=1.0@
qu.1.1.part.5.credit.1=1.0@
qu.1.1.part.6.grader=exact@
qu.1.1.part.6.editing=useHTML@
qu.1.1.part.6.answer.4=Sir Isaac Newton@
qu.1.1.part.6.question=xx@
qu.1.1.part.6.answer.3=Johannes Kepler@
qu.1.1.part.6.answer.2=Stephen Hawking@
qu.1.1.part.6.answer.1=Albert Einstein@
qu.1.1.part.6.mode=List@
qu.1.1.part.6.display=menu@
qu.1.1.part.6.credit.4=0.0@
qu.1.1.part.6.credit.3=0.0@
qu.1.1.part.6.credit.2=0.5@
qu.1.1.part.6.credit.1=1.0@
qu.1.1.algorithm=$n=int(rint(20) + 2);
$n2=$n-1;
@
qu.1.1.question=
<p>This single question demonstrates the power and flexibility of the inline question type.
There are 6 individual response cells included, each of which requires a different type of
student response.</p>

<p><i><font face="arial"  size="-1">
<b>Notes:</b>
<br>- Response cells can appear anywhere within the formatted question statement, including
within tables.
<br>- Response cells can consist of any individual question mode, including response cells that
 handle text (lenient or stringent string matching), numbers, formulas, or any other question
variety.
<br>- Partial credit can be allowed for exact-graded text or menu-style entry
cells.</font></i></p>

<p>Here is a question using a <b>symbolic formula</b> blank:
<br>Let <math><mi>y</mi><mo>=</mo><msup><mi>x</mi><mn>$n</mn></msup></math>. What is the
derivative of <math><mi>y</mi></math> with respect to <math><mi>x</mi></math>?  <1><br>
<font face="arial" size="-1"><i>Enter only the expression for the derivative, omitting "y'
=".<b>
Try it:</b>  The answer is ${mathml("$n x^($n2)")}</i> </font>
</p>

<p>Here is a <b>numeric</b> blank style of question:<br>
  3+8=<2><br>
<br><i><font face="arial"  size="-1">Numeric questions can specify margin of error and significant
 figure precision.</font></i>
</p>

<p>The following numeric blanks question requires a <b>number and a unit dimension</b> in its
response:<br>
  If a cat walks 20ft in 2min, how fast is the cat walking?  <3>
<br><i><font face="arial"  size="-1">Most standard equivalent unit dimensions are graded correctly
 by the system, or you can specify your own table of unit equivalents to apply to a
question.</p></font></i>
```

```
<p>This next question grades input in the form of a <b>text string,</b> and can also be set to
 accept multiple correct answers, so "Einstein" <i>or</i>  "einstein" will be graded as correct,
 as will "Albert Einstein"  <i>or</i> "aLbeRt einSTEIN".
<br>
  Who first developed the General Theory of Relativity? <4><br>
<br><i><i><font face="arial"  size="-1">This question <b>also</b> awards partial credit if you
 answer 'Stephen Hawking'. </font></i></i></p>

<p>This is the same question, with the grader set to accept only an <b>exact match</b> of the
answer text string, however it too can be set to accept multiple correct answers, so "Einstein"
 <i>or</i> "Albert Einstein"  will be graded as correct, but "einstein"  or  "aLbeRt einSTEIN"
 will be graded wrong.
<br>
  Who first developed the General Theory of Relativity? <5><br>
<br><i><font face="arial"  size="-1">This type of question also allows you to give some partial
 credit for certain responses, so if you enter "einstein"  or "albert einstein" you will get
1/2 credit for this question.</font></i></p>

<p>In the following question the student picks from a <b>drop-down menu</b> of choices. Partial
 credit can be set for some options, full credit for others.<br>
  Who first developed the General Theory of Relativity? <6><br>
  <font face="arial" size="-1"><i>In this case, you get full credit for 'Albert Einstein' and
1/2 credit for 'Stephen Hawking'.</font></i></p>
@
```

See Also:

Formula (page 137)

Numeric (page 146)

List Questions (page 160)

Question Designer Questions (page 166)

Short Phrase Question Example Script

The following example script creates a **Short Phrase** question.

```
qu.1.topic=Short Phrase Question Example Script@
qu.1.1.mode=Short Phrase@
qu.1.1.name=Differential geometry: Metric tensor@
qu.1.1.question=<b>Short Phrase Question</b>
<br>Which tensor is used to measure distances and angles?@
qu.1.1.answer=metric@
```

See Also:

Short Phrase Questions (page 177)

True-or-False Question Example Script

The following example script creates a **True False** question.

```
qu.1.topic=True-or-False Question Example Script@
qu.1.1.mode=True False@
qu.1.1.name=Quantum Tunneling@
qu.1.1.editing=useHTML@
qu.1.1.question=The phenomenon of quantum tunneling has a counterpart in classical physics.@
qu.1.1.answer=2@
qu.1.1.choice.1=True@
qu.1.1.choice.2=False@
```

See Also:

True-or-False Questions (page 177)

Appendix A Additional Information

A.1 Copyrights and Trademarks

Copyrights (page 303)

Trademarks (page 303)

Copyrights

- Copyright © Maplesoft, a division of Waterloo Maple Inc. 2011.

Trademarks

- Maplesoft, Maple, Waterloo Maple, and Maple T.A. are trademarks of Waterloo Maple Inc.
- Linux is a registered trademark of Linus Torvalds.
- Red Hat is a registered trademark of Red Hat, Inc.
- SUSE is a registered trademark of Novell, Inc.
- Microsoft, Excel, Internet Explorer, Windows, and Windows Vista are registered trademarks of Microsoft Corporation.
- Macintosh, Mac OS, and Safari are trademarks or registered trademarks of Apple Computer Inc.
- Firefox is a trademark of the Mozilla Foundation.
- Java, all Java based marks, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Maplesoft is independent of Sun Microsystems, Inc.
- W3C is a trademark (registered in numerous countries) of the World Wide Web Consortium; marks of W3C are registered and held by its host institutions MIT, INRIA, and Keio.
- MathType is a trademark of Design Science, Inc.
- MathML, Mathematical Markup Language is a trademark of the World Wide Web Consortium.
- Blackboard, Building Blocks, and the Blackboard Learning System are trademarks/registered trademarks of Blackboard Inc., in the United States and/or other countries. Usage by permission.
- Moodle is a registered trademark of the Moodle Trust.
- WebCT is a trademark of WebCT, Inc.

All other brand names or product names are trademarks or registered trademarks of their respective companies.

A.2 System Requirements

The following is a list of the minimum system requirements for access to the Maple T.A. system. System performance may vary based on the speed of your Internet connection. For all system systems, it is recommended that you set your screen resolution to at least 800 x 600 dpi.

Client Recommended Configuration: Instructors, Students, and Proctors

- All clients require a Java-enabled browser.
- JavaScript™ and cookies must be enabled.
- Instructors require Sun Java JRE 1.5.0 or later.
- Students require Sun Java 1.5.0 or later.

Windows - 32-bit

- Platforms: Windows XP/2003/Vista/7
- Browsers: Internet Explorer® 6.x+, Firefox® 3.0+
- 300 MHz processor or better
- 64 MB RAM or better

Windows - 64-bit

- Platforms: Windows XP/Vista/7
- Browsers: Internet Explorer 6.x+, Firefox 3.0+
- 300 MHz processor or better
- 64 MB RAM or better

Macintosh

- Platform: Mac OS X 10.5 or later
- Browsers: Firefox 3.0+, Safari 5.0+*
- 333 MHz processor or better
- 64 MB RAM or better

* The new question editor is not available on Safari. In order to use the new editor on a Macintosh, we recommend using Firefox. On Safari you can only access the authoring environment from 2.5.1.

Linux - 32-bit

- Platforms: SUSE 10.3, 11.1, 11.2; Red Hat Enterprise 5; Ubuntu 8.04, 10.04
- Browsers: Firefox 3.0+
- 300 MHz processor or better
- 64 MB RAM or better

Linux - 64-bit

- Platforms: SUSE Enterprise 10, SUSE Enterprise 11; Red Hat Enterprise 5; Ubuntu 8.04, 10.04
- Browsers: Firefox 3.0+
- 300 MHz processor or better
- 64 MB RAM or better

Internet Access

- 56K Modem Dial-up Connection
- Broadband - Cable Modem or DSL - Internet connection is recommended for optimal performance.

See Also:

Copyrights and Trademarks (page 303)

Index

Symbols

$ (dollar) Sign, 240
.qu files, 75, 116
.qu files for question banks, 283
2-D plot options, 235
2.9E8, 204
3-D plot options, 237
? operator, 127, 215, 233

A

Absolute Value Function, 204
Accessing the Assignment Editor Screen, 21
Add External Assignment, 58
Adding and Deleting Elements in a Course Module, 66
Adding assignment, 23
Adding Questions in Assignment Editor
  Add individual questions, 38
  Add questions as a group, 39
Advanced button
  Assignment Name tab, 38
  Set Assignment Policies tab, 46
  Text on Results Page field, 38
Advanced Question Types
  Applets, 178
Algebraic Symbols, 130
Algorithm, 233, 241
Algorithm authoring, 218, 229
Algorithm Editor, 219
Algorithmic Question Capabilities, 217
Algorithmic variables
  Details evaluation, 239
Algorithmically generated, 78, 219
  Formula question, 284
  Multiple selection question, 284
Algorithmically generated Variables, 245
Algorithms, 78, 122, 176, 219, 221, 243, 265
  Defined in the Algorithm Designer, 80
allow2d, 143
Annotations
  Creating an assignment annotation, 40
  Uses, 40
Anonymous practice, 35
Answer Field, 127
Answer field evaluation, 239
Answer Region, 170
Answers, 85, 122, 125, 127, 184, 213, 215, 233, 239
  Defining correct responses, 181
API, 178
API for Questions, 288
Applet, 89, 124
  plotting, 259
Applet Questions, 151
  Authoring, 288
Applets.grapher.Graph, 297
Arccos, 180, 233
Arcsin, 180, 233
Arctan, 180, 233
Assigning partial credit, 27
Assignment
  set prerequisites, 46
Assignment Editor, 27, 28
  Choose Name tab, 37
  Details, 47
  Options, 23
  Review & Finish tab, 32
  Select Questions tab, 28
  Set Policies tab, 30
Assignment Editor Main Menu
  Add button, 23
  Backup button, 47
  Delete button, 24
  Move Up Move Down buttons, 26
  Print button, 26
  Shared content icon, 27
Assignment grades, 51
Assignment name
  Choose Name tab, 28
  Student Menu, 37
Assignment Options
  Feedback, 43
  Final grade, 43
  Hints during assignment, 43
  Send email reports, 43
Assignment properties
  Passing score, 44
  Questions per page, 44
  Time limit, 44
Assignment type
  Anonymous practice, 35
  Homework or Quiz, 34
  Mastery, 36
  Proctored exam, 35
  Setting, 42
  Study session, 37
Assignment types, 35
Assignment visibility, 46
Assignments, 1
  Deleting questions, 39
  Forced grading, 27, 59
  Header, 38
  List order, 26
  Reviewing Details, 47
Author question banks, 247

Author Questions, 73
  Using Script Files, 283
Authoring, 73
  Applet Questions, 288
  Graphing, 297
  Mathematical, 179
  multipart, 100
  Pathways, 92
Authoring Path, 75
Authoring with Question Designer, 167
Axes
  Display in Interactive Plots, 90

B

Backing up questions, 116, 120
Backslash symbol
  Displaying, 240
Basic Math Syntax, 180
Behaviors, 169, 171
Blanks, 124, 133, 289
Blanks Questions, 134, 158, 250, 253
  and Short Phrase Questions, 159
  [formula] option, 268
Browser, 73

C

C, 205
Canceling assignment, 47
Case-sensitivity, 158
Changing grades, 53
Checkmark icon, 170
Chemistry, 135
Child classes
  Gradebook, 53
Choose Name tab, 23
  Advanced button, 38
  Assignment name, 28
Class Homepage, 1
  Guidelines, 4
Class Name, 9
Class Registration
  Locking and unlocking, 10
Class roster
  Creating and uploading, 11
  File type, 11
Class web site
  Flash movie, 89
  Images and reference files in questions, 87
  Uploading images and references files, 87
  Uploading multiple files, 18
  Uploading single files, 18
  Viewing or deleting files, 18
  Working with folders, 19
Clickable, 252
Clickable Image Question, 87, 154
  Example script, 288
Clickable Image Questions, 93
Clickable imagemap, 124
Clone Question, 114
Cloned Questions, 117
Coaching, 128
Comma-delimited, 58
commas, 139
Comment
  Deleting, 84
Comment Field, 129
Comment Field for Maple-graded, 80, 96, 140
Comments, 78, 122, 213
Condition, 222
Constant, 132, 136, 269
Content Updates, 13
Controlling Answer Format, 182
Coordinates, 280
COPY
  From Maple Worksheet, 96
Copying Assignments, 23
Copying questions, 77
Copyright, 303
Cos, 233
Course ID, 9
Course Module
  Creating, 63
  Deleting, 67
  Exporting, 64
  Importing and installing, 65
  Relative web references, 67
Course Modules
  Viewing Details, 66
Course Modules Overview, 61
Create
  Course Module, 63
  multipart, 100
  New Class, 7
  Shared Class, 14
Create Multiple Selection Questions
  Using Scripts, 296
Creating a Course Module, 63
Creating Pop-Up References, 105
Csc, 233

D

d-Value, 55
Decimal, 133, 214, 223
Decimal places, 239
Define
  Answer Region, 170

Response Area, 167
Define grading, 171
Define Response Area, 167
Deg, 205
Delete Question, 115
Delete Roster from Class, 11
Deleted Gradebook results, 24
Deleting
Assignment, 24
Course Module, 67
Files using Web Site Editor, 18
Proctors, 17
Questions from assignments, 39
Student record, 69
Deleting a Course Module, 67
Description URL, 9
Different Forms, 132
Digits in Maple Questions, 185
Dimensioned Formula, 124, 132, 139
Dimensioned Number, 124
Disable vs Delete assignment, 24
Display
Answers vs Formulas, 213
Displaying
$, 240
Backslash symbol, 240
Displaying a Plot, 201
drawMaplePlot, 201
Drop-down List Questions, 153
Duplicating
Questions, 77

E

E
Maple questions, 143
e, 204
Naming variables, 221
Ed.sty, 247
Edit
HTML, 168
Edit existing questions, 75
Edit Profile, 69
Edit Question, 114
Edit Response Area, 170
Editing
Questions, 76, 84
Source, 84
Editing assignments, 25
Editing with the Equation Editor, 188
Editor, 168
Electromagnetism, 205
Email notification
Passing score, 43
Empty Class, 7
Entering Chemistry Expressions, 186
Equals condition, 223
Equation, 124, 132, 137
Equation Editor, 168
Equation Editor Palette, 190
Equation Editor Requirements and Syntax, 192
Equation Question Types, 179
Equation Questions, 179, 270
Equivalent
Unit dimensions, 139
Equivalent Answers, 127
Equivalent Units
Creating New Table, 211
Erf, 231
Error
Margin, 184
Essay Question, 171
Essay Question Example Script, 289
Essay questions
Grading, 53
Essay Questions, 155, 289
Creating in the Question Editor, 94
Euler's Constant, 180
Example Script
Library of Example Plain text script files, 301
Maple Formula, 293
Maple Syntax, 294
Excel, 58
Export Course Module, 64
Export Question Bank File, 114
Exporting
Grades, 58
External Assignment, 58
Extra credit
Gradebook, 57
e^x, 180

F

F, 205
Factorial, 224
Featured Class, 9
Features, xvii
Feedback, 30, 43, 128
Feedback for Maple-graded questions, 140
Feedback Options
Configuring, 129
Field Type, 122
Figure Labels, 284
Fill-in-the-blank, 95, 153, 158, 159, 169
create, 289
Fill-in-the-Blank Question Example Script, 289
Fill-in-the-Blank Questions, 153, 159

Final grade, 43
Finishing assignments, 47
Flash movie, 89
Flash Questions, 155
Folders, 19
Force grade, 59
Formatting, 168
Formula, 171, 179, 188, 213, 219, 228, 233, 239, 284
  Blanks, 134
  Blanks in LaTeX, 268
  in Question Editor, 98
  Multipart, 146
  Restricted, 150
  Restricted in LaTeX, 279
  Unordered List, 145
  Vectors, 150
  with unit dimensions, 139
Formula - Form Variant, 138
Formula List, 124, 139, 271, 280
Formula list Questions, 271
Formula Mod C, 124, 136, 269
Formula Questions, 132, 137, 271
  Multipart, 146
FormulaList, 271
frac, 224
Free Response, 95, 159
Functions within Algorithms, 219

G

G, 205
gcd, 225
Getting help, 5
Getting Started, 2
GIF, 87, 93, 243
Gif file
  Including in LaTeX Questions, 262
  Labeling in LaTeX, 263
Global proctor, 35
Grade, 85
Grade in course
  computation, 57
Grade reports, 57
Gradebook, 1
  Add External Assignment, 58
  Assignment grades, 51
  Class grades, 53
  Deleted results, 24
  Item statistics, 55
  Overview, 49
  Search, 50
  Security, 69
  snapshots, 57
  Student statistics, 52
Graded Free-response, 179
Grades
  Changing, 53
Grading
  Essay questions, 53
  Free Response Text Questions, 158
  Math Input, 181
  Multiple selection question, 164
  Testing, 85
Grading policy
  Grade report, 57
Graph Plotting Applet, 149
Graph Plotting in Plain Text, 290
Graph Sketching Question Example Script, 290
Graph-sketching, 260
Graphic Labels, 284
Graphic.eps, 262
Graphic.gif, 262
Graphic.jpg, 262
Graphic.pct, 262
Graphics
  Creating pop-ups, 105
  Including, 262
  Labeling, 263
Graphics Elements, 259
Graphing, 148, 260
Graphing Questions
  Authoring, 297
Greater than (condition), 225
Greek Letters, 130
Group Questions in the Question Repository, 115
Gt, 225

H

Hide assignment, 24, 46
Hide assignment from view, 24
Hints, 83, 122, 128, 129
  Deleting, 84
Hints during assignment, 43
Homework or Quiz, 34
HTML, 130, 168
HTML Formatting, 107
Hz, 205

I

if, 225
Image, 87
Image files, 73
Images, 93, 154, 167, 252, 262
  file names, 87
Img src, 87
Importing and Installing Course Module, 65
Indefinite Integrals, 136

indexof, 225
Info Field, 82, 128
Information, 82
Inherit Content From, 9
Inherited Questions, 117
Inline Display of Formatted Math Expressions, 196
Inline Variable Expressions, 220
Insert
  Algorithm, 176
  Response Area, 170
Insert Algorithm, 176
Insert/Edit Response Area, 170
Inserting
  MathML, 168
Install Bank, 121
Instructions & Links
  Question annotations, 40
Instructor
  Changing your password, 69
  Login, 3
  Name, 9
  Password security, 69
Instructor Help, xvii
  Advanced, 73
Integer, 223, 228, 239
  Random, 230
Integers
  Formatting, 214
Integration, 124, 132, 136
Integration Questions
  Constants, 269
Internal Directive, 122
Internet Access, 303
Inverse Cosine, 204
Inverse Sine, 204
Inverse Tangent, 204
Item statistics, 55

J

J, 205
java, 225
Java applet, 178
JPEG, 87

K

Kelvin, 205
Key Word, 160, 291
KHz, 205
KiloHertz, 205
KiloJoules, 205
KiloWatts, 205
KJ, 205
KW, 205

L

Labelgraphic, 263
Labeling
  Graphics, 263
Labeling Graphics in Questions Example Script, 291
LaTeX, 75, 144, 159, 247, 297
  Maple Questions, 272
  question fields, 248
  Question Structure, 248
  Question types, 257
  Question Using Algorithms, 265
LaTeX Authoring Overview, 247
LaTeX Question Bank Document Structure, 247
LaTeX Style Sheet Tips, 249
LDAP, 71
  Add user, 72
  Import, 72
Least significant unit, 183, 226
Linux, 303
List, 160
List Question, 171
List Question Example Script, 292
List-based variables, 232, 240, 243
Lm, 205
Ln, 233
Load, 120
Local proctor, 35
Locked assignment icon, 27
Locking class registration, 10
Logical Operators, 233
Login, 1
  Instructor, 3
  Using university logins, 71
Lsu, 183, 226
Lux, 205
Lx, 205

M

Macintosh, 303
Managing Proctors, 15
Maple, 187, 201, 203, 227
Maple Question, 171
Maple Questions, 140
  $RESPONSE variable, 96
  Boolean value, 96
  Digits, 185
  e, 142, 143, 221
  Feedback in Basic Question, 294
  in the Question Editor, 96
  LaTeX, 272
  Linking to a Maple Repository, 90
  Maple Formula Example Script, 293
  Maple Syntax, 143

Maple Syntax Example Script, 294
Maple-graded Formula Questions, 142
Partial Grading, 96
Preventing Cheating, 185
Unevaluated forms, 143
Using a Maple Repository, 90
Maple Repository, 90
Maple Syntax, 143
Preventing Cheating, 185
Maple Syntax Questions
Editing, 107
Maple T.A.
Using in the classroom, 5
maple(text), 226
Margin of error, 86, 181, 183, 184, 215
Margin of tolerance, 86, 183, 184, 215
Mastery assignments, 36
Edit mastery policies link, 43
Setting policies, 43
Study Session and Mastery, 36
Matching, 161, 254
in Question Editor, 98
Math capabilities, 181
Math Expressions, 168
Authoring, 198
From the Equation Editor, 195
Using MathML, 199
Math Functions, 204
Math grading, 169
Math Input
Grading, 181
Math Question Types, 149
Math Question Types Comparison Table, 133
Math Questions, 133, 184
Math Syntax in Questions, 180
Math/formula, 132
Mathematical Fill-in-the-Blanks, 268
Mathematical Formula Free Response, 169
Mathematical Formula Question, 284
Mathematical Formula Questions, 98
Mathematical Functions, 233
Mathematical Questions
Authoring, 179
Mathematics Question Types, 179
Mathematics Questions, 179
Answers, 239
MathML, 122, 130, 199, 271, 276
Mathml, 168
MathML Authoring, 197
MathML support, 75
mathml(f), 226
MathType, 197
Matrices
Create Maple question using palette, 107
Matrix, 124, 132, 144
Matrix palette, 107
Matrix Questions, 274
max, 227
Merging
questions into question groups, 39
Metadata, 122, 128
MiKTeX, 249
Min, 205
min, 227
Misc, 205
Mol, 205
Move Up Move Down, 26
Movie clips, 73
Multi Formula, 132, 145
Multiformula Questions, 274
Multipart, 124, 137, 146, 163
authoring, 100
Multipart Questions, 100, 162, 163, 255, 295
Editing, 112
Multiple Choice, 101, 124, 169, 171
Multiple Choice Question Example Script, 296
Multiple Choice Questions, 163, 256
Multiple Selection, 124, 169
Multiple Selection Question Example Script, 296
Multiple Selection Questions, 101, 164, 257, 284, 296
Multiple-choice, 163, 177
My Profile, 69

N

Name, 122, 126
Natural Logarithm, 204
New Class
Create, 7
New Question
Create, 76
Nm, 205
Nonpermuting Multiple Selection, 257
NOT Celsius, 205
not equal to (condition), 227
Not equals condition, 223
not(a), 227
Ns, 205
Nth, 226
Nth item, 232
Ntuple, 124, 133, 150, 280
Ntuples, 274
Number, 124, 132, 146, 179, 184, 221
Number following, 276
Numbers within Questions, 239
Numeric, 146
in Question Editor, 102
Numeric Free Response, 169

Numeric Question, 171
Numeric Question Types, 179
Numeric Questions, 179, 215, 276
  in Question Editor, 102
Numeric/formula, 124
numfmt, 227

O

Online Authoring, 92
Online Help, 5
Open Response Text-based Questions, 253
Operations, 233
Order
  Assignment list, 26
Ordered, 124, 133, 139
Overview, xvii, 1
  Course Modules, 61

P

p-Biserial, 55
p-Value, 55
Palette Questions, 165
  in Question Editor, 103
Palettes, 168
Partial credit, 27
  Maple-graded questions, 97
  Multipart Questions, 113
  Tutorial Multipart Questions, 112
Partial Credit
  Multipart Questions, 163
Partial marks, 27
Pascals, 205
Password, 69
  Forgotten, 71
Penalties
  in Mastery assignments, 43
Performance data, 58
Performance Feedback, 128, 129
Phrase questions, 160, 291
Physical Unit Equivalents
  Table, 205
Physics/engineering, 183
Plain Number, 124, 146
Plain text question bank files, 75, 283
Planning
  Your Question Bank Project, 75
Platforms, 73
plotmaple, 203, 227
Plotting, 148, 203, 227
  applet, 259
  Displaying a Plot, 201
  Student Response, 187
plotting
  2-D options, 235
  3-D options, 237
PNG, 87
Points column, 42
Pop-up HTML reference, 105
Precision, 86, 183
Previewing
  Assignments, 26
  Questions for selection, 39
  Questions in assignments, 26, 40
Printing assignments, 26
Proctor
  Adding, 16
  Deleting, 17
  Managing, 15
Proctor Overview, 15
Proctored exams
  Proctor types, 35
Profile
  Edit, 69
Properties, 169
Public Questions, 117

Q

Qu.x.topic, 118
Qu.x.y.algorithm, 219
Qu.x.y.answer, 213
Qu.x.y.comment, 213
Qu.x.y.mode, 124, 137, 145
Qu.x.y.name, 126
Qu.x.y.solution, 129
Question
  Adding, 76
  Deleting, 115
  Duplicating, 77
  Essay, 169, 171
  Formula, 169, 171
  List, 169, 171
  Maple, 169, 171
  Multiple Choice, 169, 171
  Numeric, 169, 171
  Permanently Deleting, 115
  Previewing, 116
Question Annotations, 40
Question authoring, 95
Question Bank File
  Exporting, 114
  Opening, 120
Question Bank File format: Plain text, 75
Question bank file formats
  Plain text script files, 283
Question bank files, 73
Question Banks, 1, 75, 116, 125, 129

Authoring, 73
Open, 120
Reinstalling, 121
Question Chaining Variables, 233
Question Descriptions, 126
Question Designer, 167
Question Designer Example with List Questions, 298
Question Designer Questions, 175
Question Designer Questions in the Question Editor, 104
Question Editor, 75, 92, 113
Essay, 94
Pop-Ups, 105
True-or-False, 105
Question Fields, 122
Question Mode, 124, 132, 137, 155, 162, 177, 254, 269, 271
Redefining, 85
Question Modes, 131, 133, 134, 139, 163, 250
Question parts, 163
Sequencing Parts, 163
Question Repository, 115
Clean up, 115
Question Statement, 125
Question Types, 95, 124, 131, 169, 274, 289
Drop-down List, 153
Equation, 137
Fill-in-the-Blanks, 153, 159
Free Response, 159
in the Question Editor, 113
Key Word, 160
List, 160
Maple-graded, 140
Math, 149
Matrix, 144
Multipart, 162
Short Phrase, 177
Question weighting, 42, 112
Questions, 233
save to hard drive, 116, 120
Sharing, 117
Questions menu, 73
Questions with Images, 106
Questions with Interactive Plots, 90

R

r-Biserial, 55
Rad, 205
Random Integer, 221, 265
Random Integer Generation, 230
Random number generation, 221, 228, 229
Randomized Formula, 124, 297
Range, 229
rank, 230
Reals, 214
Redefining
Question Types, 85
Reference Files, 87
Reference Sources, 87
Register students
in a class, 10
using class rosters, 11
using the User manager, 12
Registration
Locked, 9
Registration code, 14
Reinstalling
Question Bank, 121
Relative Web References in Course Modules, 67
Release, xvii
Reports, 1
d-Value, 55
Exporting, 58
File types, 58
p-Value, 55
Required Question Elements, 125
Resequence questions, 163
Resequence topics, 163
Response Area Properties, 171
Response Areas, 167
Response-specific, 101, 122, 129, 163
Hints, 128
Restricted Formula, 124, 150
Restricted Formula Questions, 279
Review & Finish tab, 32
Canceling assignment, 47
Finishing and Saving assignments, 47
Reviewing assignment details, 47
Reviewing
Assignment details, 47
Rint, 221, 228, 230
Roster, 11
Roster add
External assignment, 58
Rules, 158, 188
Mastery assignments, 36
Naming variables, 221

S

S, 205
Save
back up questions to hard drive, 116, 120
Saved Question Bank File
Opening, 120
Saving, 121
Saving assignments, 47
Saving shared assignments, 25

Saving table from Gradebook, 58
Scenario-based variables, 243
Scheduling assignments, 45
School Name, 9
Scientific notation, 132, 146, 179, 180, 276
Scramble questions, 42
Script file authoring, 283
Script Files, 73, 288, 295, 301
Search for Questions, 114
Security
  Proctors, 69
  Student account management, 69
Select Questions tab, 38
  Add questions as group, 39
  Add questions to assignments, 38
  Annotations, 40
  Merging questions into question groups, 39
  Points column, 42
  Scramble questions, 42
Select Test Bank, 38, 39
Select Test Bank in Assignment Editor, 28
Self-registration, 10
semicolons, 145
Send email reports, 43
Sequence of listed assignments, 26
Sequencing Parts, 163
Set Policies tab, 30
  Advanced button, 46
  Assignment options, 43
  Feedback options, 44
  Scheduling assignments, 45
  Set assignment type, 42
SetImageBase, 262
Setting Answer Format in Numeric Questions, 85
Shadowed Questions, 117
Share Question, 114
Shared assignments
  Saving, 25
  Shared content icon, 27
Shared class
  Behavior, 13
  Create, 7
  Creating, 14
Shared Class Content Updates, 13
Shared Class Overview, 13
Shared content icon, 13, 27
Sharing questions, 117
Short Answer Questions, 177, 300
Short Phrase, 124, 177, 300
Sig, 214, 239
Sigma, 168
Significant Digits, 185, 226
  Controlling, 183
Significant figures, 223
Simplified Math Expressions, 199
Solution, 83, 122
  Deleting, 84
Solution field, 129
Sound files, 73
Special Characters, 130
Sqrt, 180, 233
Square Root, 204
Statistical Operations, 231
strcat, 232
Student
  Changing grades, 53
  Deleting student record, 69
Student Menu
  Assignment name, 37
Student Response
  Plotting, 187
Student result
  Adding comments, 53
StudentResponses
  Grading, 187
Study Session Assignments, 37, 128, 129
  Study Session and Mastery, 36
Success rate, 55
Sum, 232
Support, 5
Switch, 232, 243
Symbol mode
  in student response, 143, 182, 187
System, 1
System Homepage, 4
System Requirements, 303
System Time-outs During Authoring Sessions, 77

T

Tab-delimited, 58
Table
  Assignment types overview, 33
  Physical Unit Equivalents, 205
Telsa, 205
Test Bank in Assignment Editor, 28
  Add questions as a group to assignments, 39
  Adding individual questions to assignments, 38
  Previewing questions for selection, 39
Testbank, 120
  Installation, 120
Testbank.qu, 116
Testing
  Grading, 85
Text Formatting within Questions, 107
Text mode
  in student response, 143, 182, 187
Text strings within variable statements, 240

Time-outs During Authoring Sessions, 77
Tips
  Formatting, 168
Tolerance, 86, 122, 146, 183, 184, 215, 226, 285
Topic, 122, 125, 129
  Open, 121
Topic Structure within Question Banks, 118
Trademark, 303
Trig Functions - restricting entry, 279
Trigonometric Functions, 204
True-or-False Questions, 124, 177
  Creating, 105
  LaTeX, 258
  Script file example, 300
Tutorial
  Algorithmic question, 241
  Basic Gradebook Searches, 50
  Formatting integers, 214
  List-based variables, 243
  Maple-based algorithmic variables, 245
  Margin of tolerance, 215
  Matrix questions, 107
  Multipart questions, 112
  Providing answers, 213
  Using the Assignment Editor, 26
Types Comparison Table, 132

U

Unevaluated forms, 143, 204
Ungraded Essay Questions, 155
Unicode HTML, 130
Units, 132, 139, 146, 179, 183, 184
  Equivalents, 205
Unlocking class registration, 10
Unordered List, 124, 133
  Formulas, 145
Uploading
  individual reference files, 87
  multiple reference files, 87
  Saved Question Bank File, 120
Uploading multiple files, 18
Uploading single files, 18
URL
  Class Homepage, 4
Use
  HTML, 168
User manager
  register students, 12
  Registering students, 10
User Manager
  LDAP, 72
User Roles, 1
Username, 71
Using /epsfig, 262
Using Algorithms, 265
Using Formula Syntax, 188
Using Hints, 128
Using Images
  File naming rules, 87
Using Information Fields, 128
Using Script Files
  Author Questions, 283
Using Scripts, 291
  Create Multipart Questions, 295
  Create Multiple Choice Questions, 296
  Create Multiple Selection Questions, 296
  Create Short Answer Questions, 300
  Feedback in Maple Question, 294

V

Variable, 133, 134, 180, 217, 219, 228, 233
  Display, 239
  Names, 221
  Names with braces, 213
Variable Data, 219
  Algorithmic, 284
  In Algorithmic Questions, 218
  Tutorial Algorithmic Questions, 241
  With range, 229
Variable group, 243
Variables
  Conditions, 223
  e, 221
  In LaTeX, 265
  List-based, 240
  Lists of variables, 232
  Setting conditions, 222
Variables in Questions, 229
Variables within variable statements, 239
Vectors, 124, 132
  Formulas, 150
Vectors Questions, 280
Versioning
  Scramble questions, 42
Viewing Details of a Course Module, 66
Viewing files, 18
Visible
  assignment property, 46

W

Wb, 205
Web files, 73
Weber, 205
Weighting
  Question, 42
What's New, xvii

Windows, 303
Worked Solution, 83
WYSIWYG, 168

Y

Your Question Bank Project
    Planning, 75